EMPIRE CODE

DO NOT DELETE

Guiding notes for editor:

- Supervisor advises on total hours for syllabus and project scale.
- Each point should be 10-15 words max.
- No changes to format (text style, colors, tables).
- Replace blue text only.
- Design tasks to be modular and independent, so each task functions on its own.
- Copy/paste rows/tables where relevant (e.g., project table for more projects).
- Each session is one class; some cover two lessons based on student availability.

Tasks Assigned by Supervisor **Note:* • *Self-initiate and work on the next project once initial tasks are done.* • *Check in regularly with the supervisor for clarifications.* • *Align expectations at each step when unsure.*	**Deadlines**

EMPIRE CODE

MIT APP INVENTOR

Learning Objectives

1. **Learning Objective 1:**
   Integrate AI Functionality into App Development:
   Students will learn to design and implement AI-powered features, such as chatbots, image generators, and input filtering systems, to create apps with dynamic and purposeful functionality.

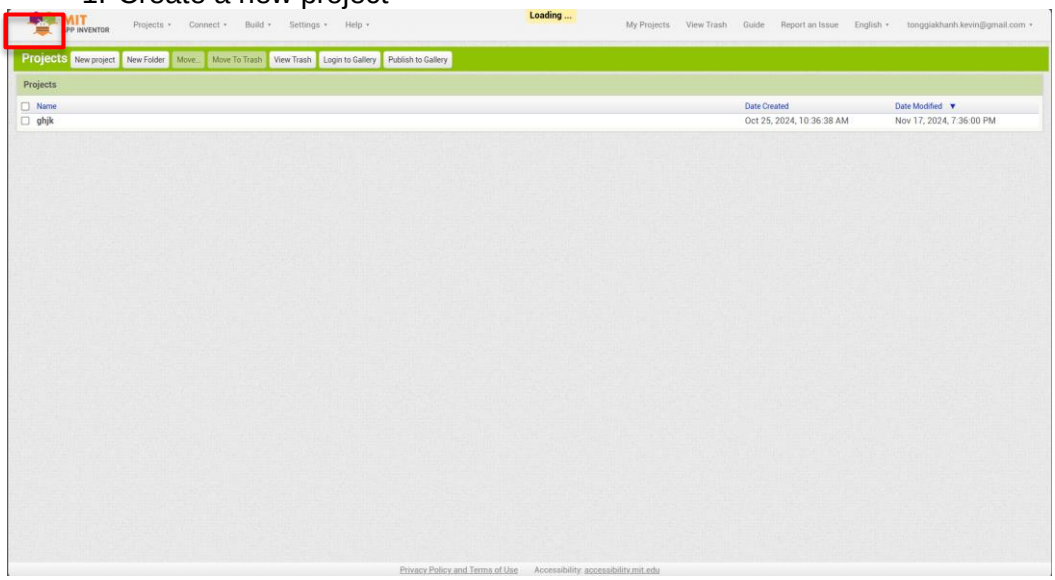
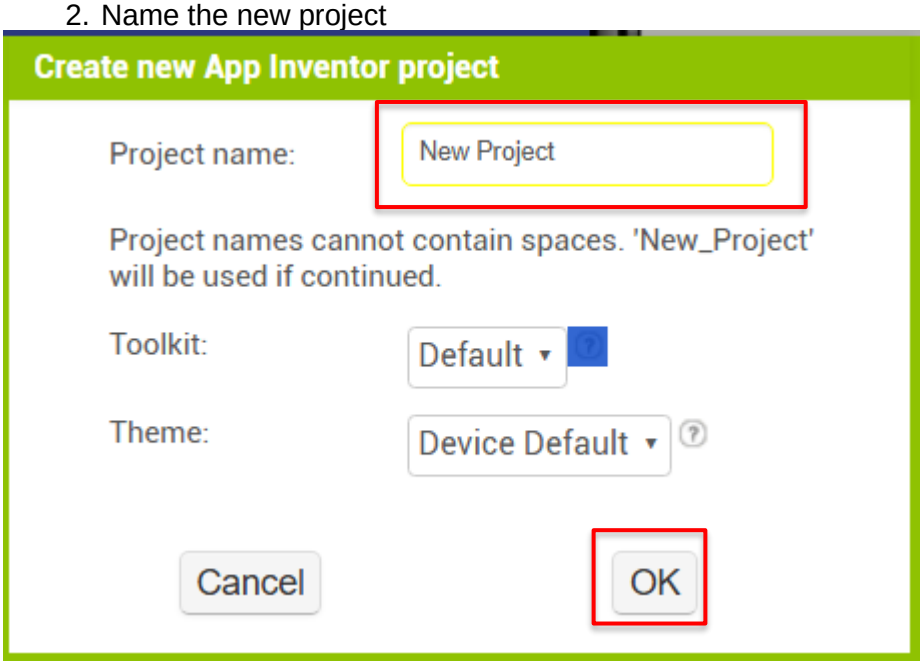
2. **Learning Objective 2:**
   Develop Problem-Solving and Logical Thinking Skills:
   Students will apply programming concepts such as list management, conditionals, and iterative logic to create user-friendly applications that align with specific themes, such as sustainability and environmental advocacy.

3. **Learning Objective 3:**
   Enhance User Experience Through Interface and Backend Integration:
   Students will design intuitive app interfaces while implementing robust backend logic to process user inputs, validate data, and display relevant outputs effectively.

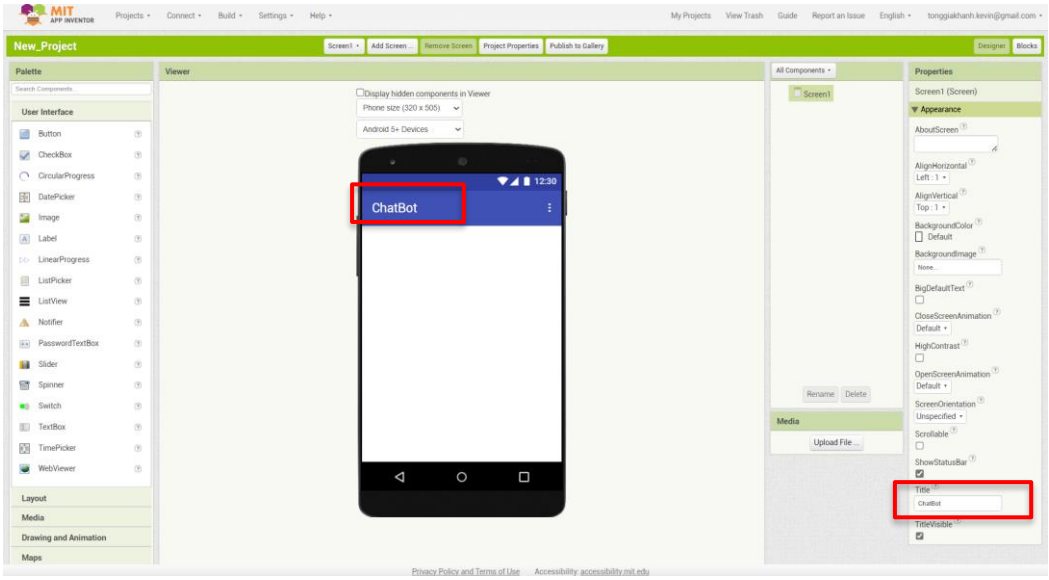
Syllabus (20hrs)

- **Project 1 – AI Chat Bot**
- **Project 2 – AI Image Generator**
- **Project 3 – AI Input filtering system**

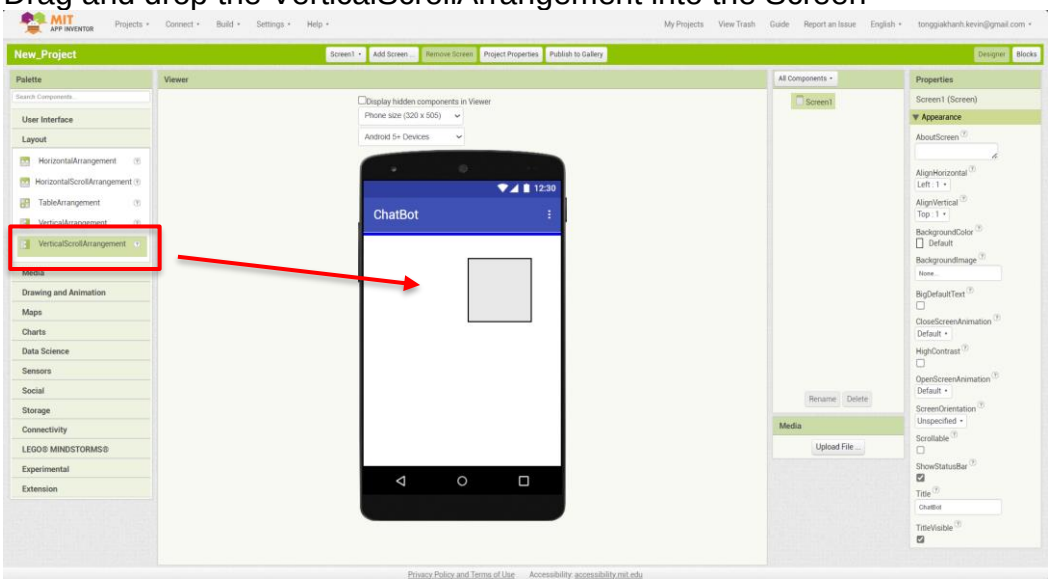
Project 1 – AI ChatBot (Kevin) (hrs)	
Learning Outcomes (5mins)	**Learning Outcome 1: Learn to Integrate AI Chatbot Components into App Development** ● Students will understand how to incorporate an AI chatbot as a functional component within an app, combining frontend design with backend logic. **Learning Outcome 2: Develop Full-Stack Application Skills** ● Students will build an interactive user interface to facilitate chatbot communication. ● Implement backend functionality to handle user input, generate AI responses, and display conversations. **Learning Outcome 3: Understand and Utilize Key Functions and Variables in App Development:** ● Students will learn to use essential programming constructs such as function calls, commands, and variables to enable seamless data transfer between the chatbot and the app interface.

Session No.: **Session Title** (hrs)	**Project Brief** ● This project guides students in creating an AI Chatbot using MIT App Inventor, covering both interface design and backend development. **Task:** 1. Create a new project  2. Name the new project 

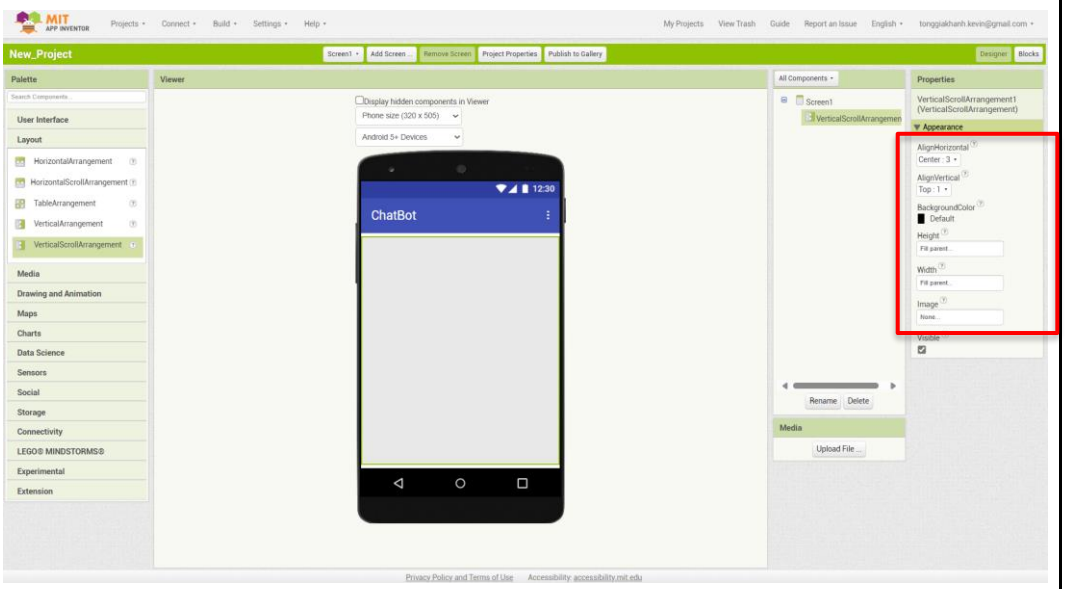
Rename the Screen



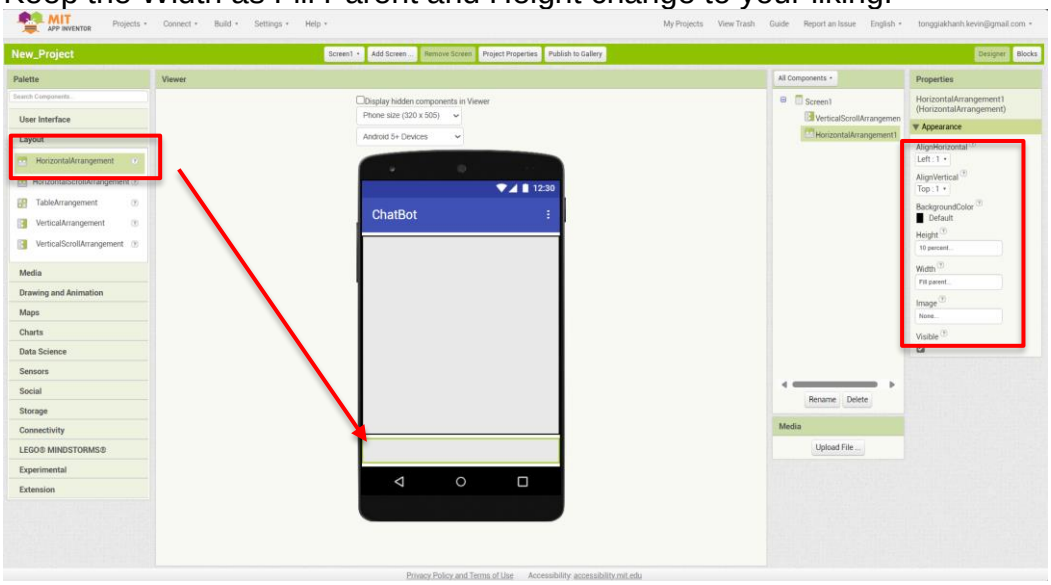
Drag and drop the VerticalScrollArrangement into the Screen



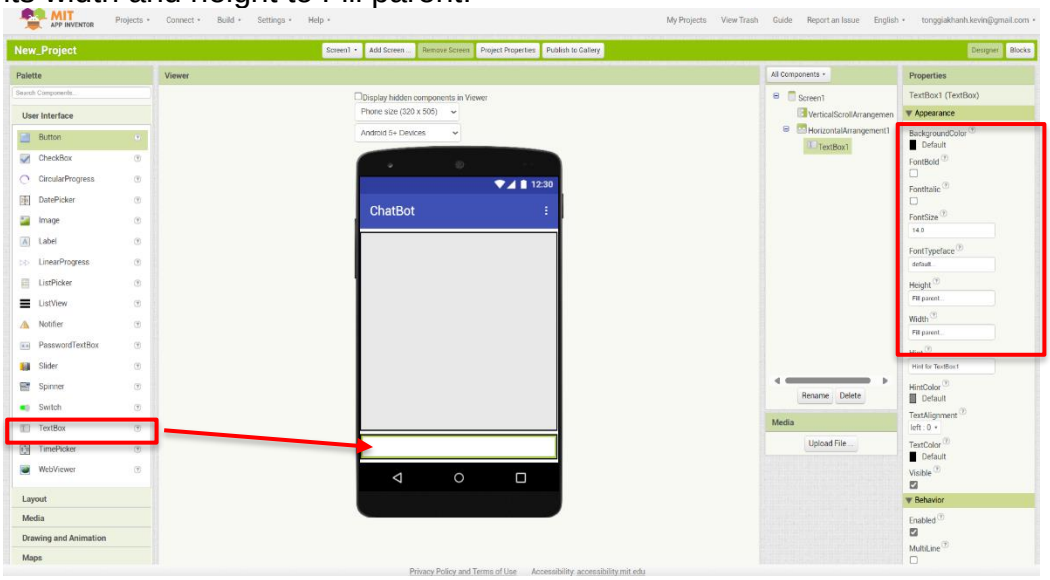
Choose “Center” for AlignHorizontal and “Fill Parent” for both Width and Height. This fills the whole screen with the arrangement but still allows the other components to be dynamically added in.



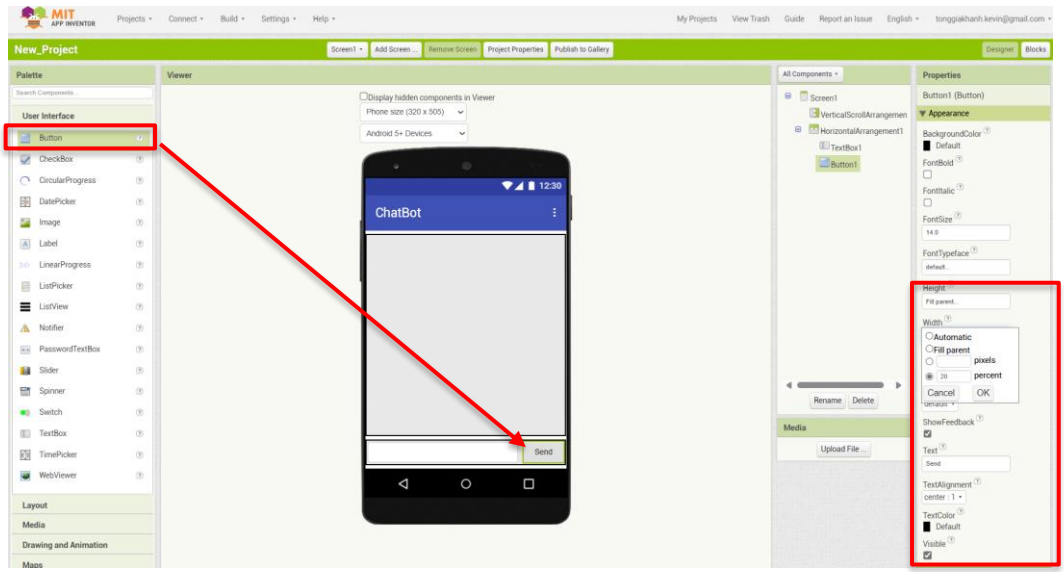
Drag and drop the Horizontal Arrangement for the underneath the VerticalScrollArrangement for the textbox and send button to be placed later.
Keep the Width as Fill Parent and Height change to your liking.



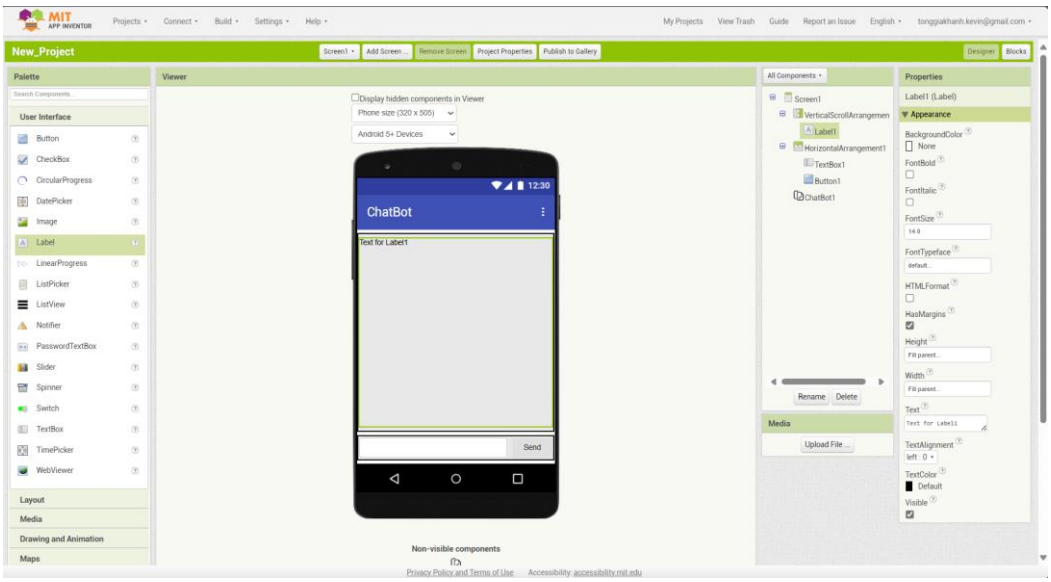
Drag & drop the text box component into the HorizontalArrangement. Set its width and height to Fill parent.



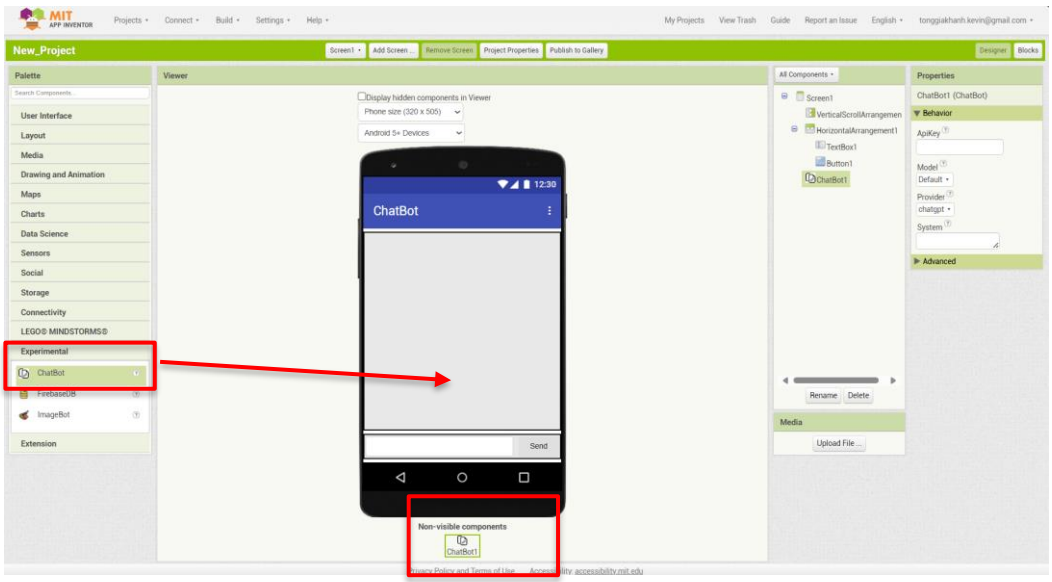
Drag & drop the button component into the HorizontalArrangement. Set its height to Fill parent and its width to your liking. Change the text inside of the button.



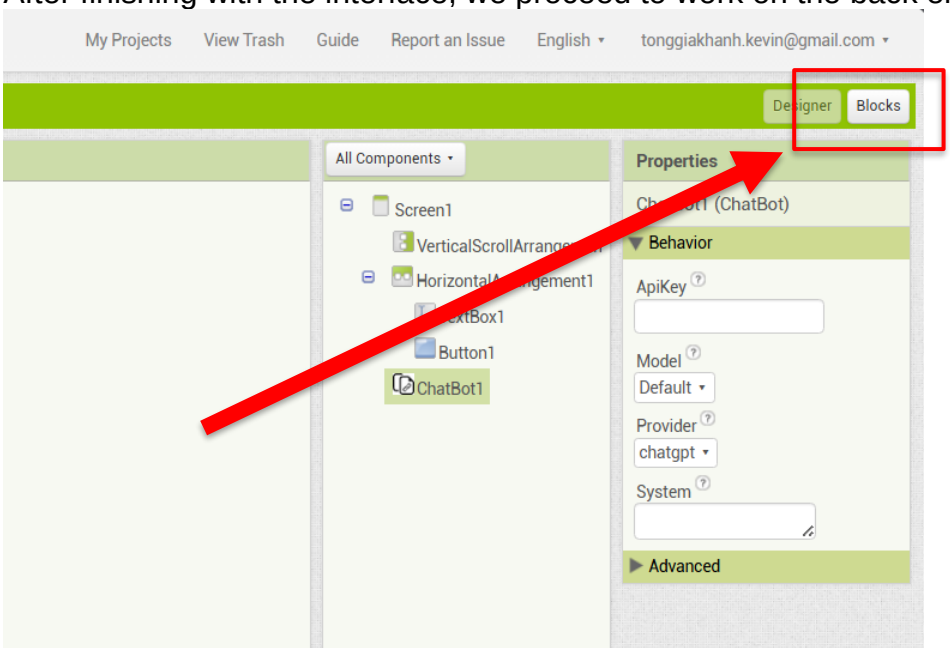
Drag the label component



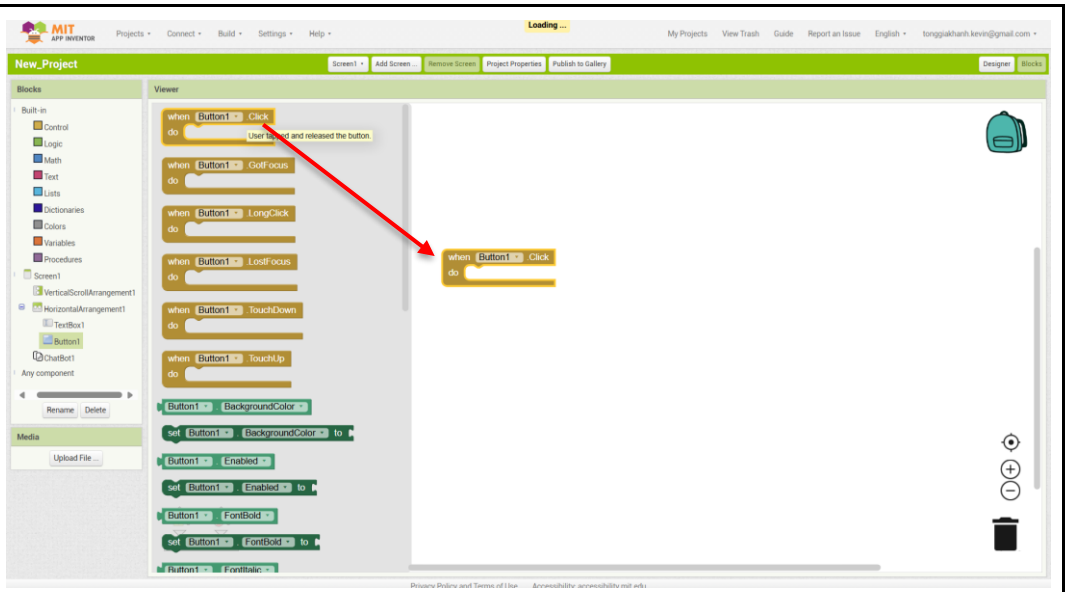
Drag & drop the ChatBot component under Experimental tab into the screen. The component will appear below the device as it is a non-visible component

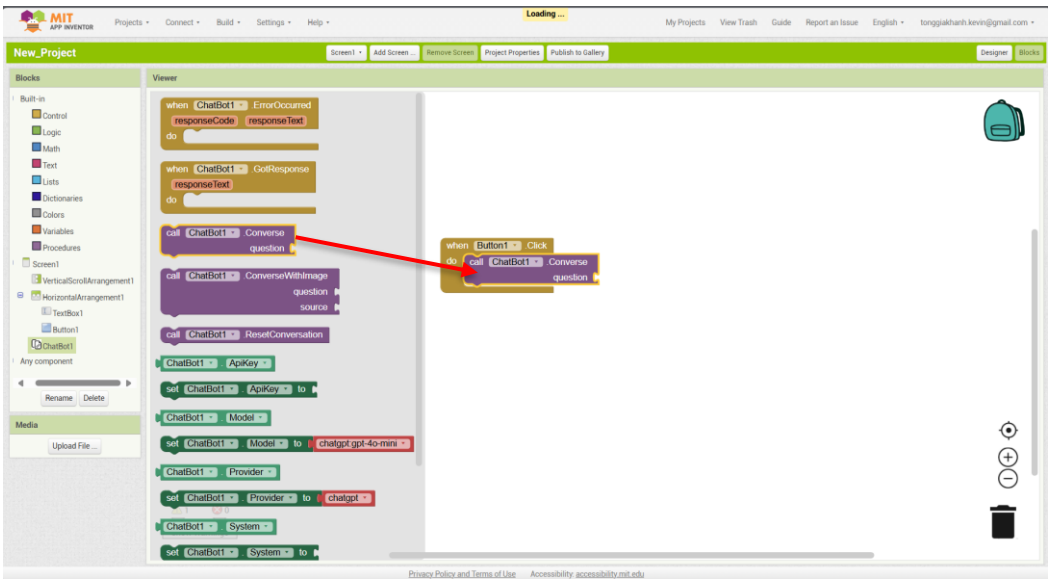


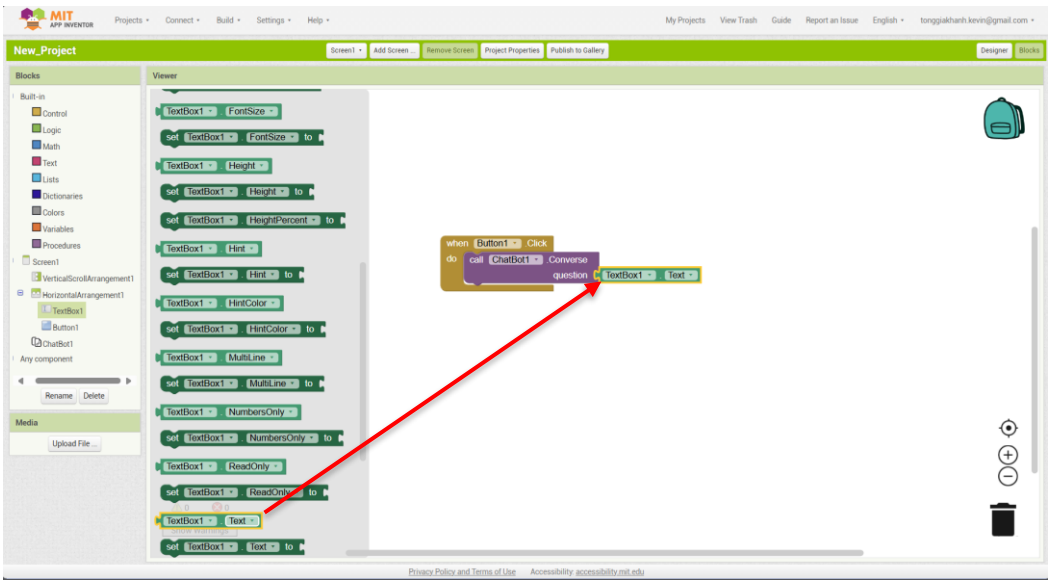
After finishing with the interface, we proceed to work on the back end

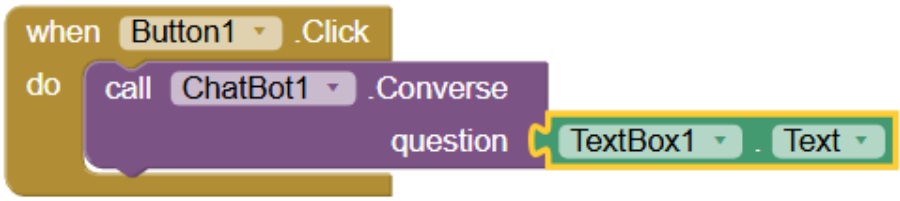


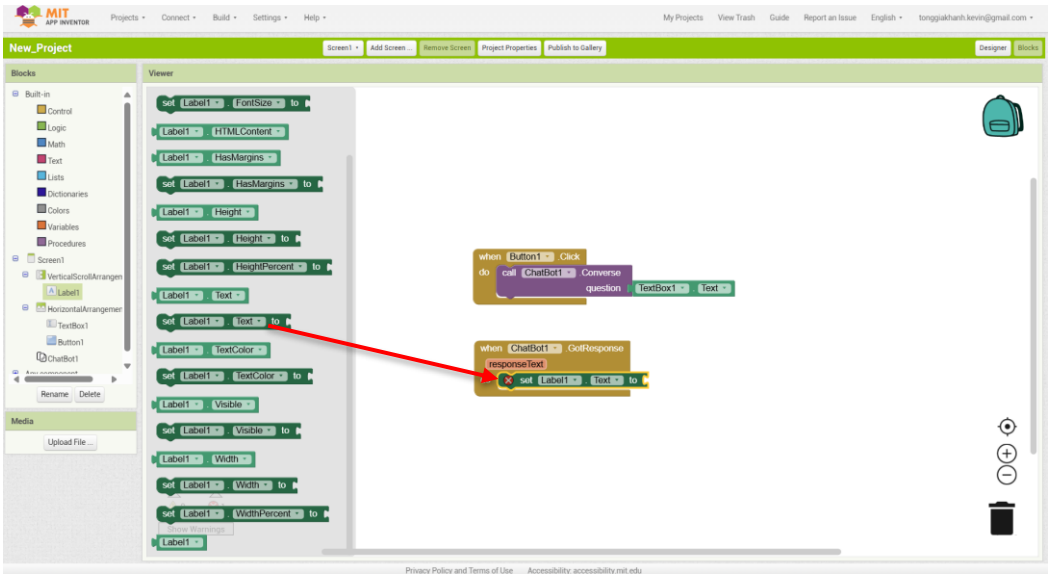
Drag Button1.Click into the space

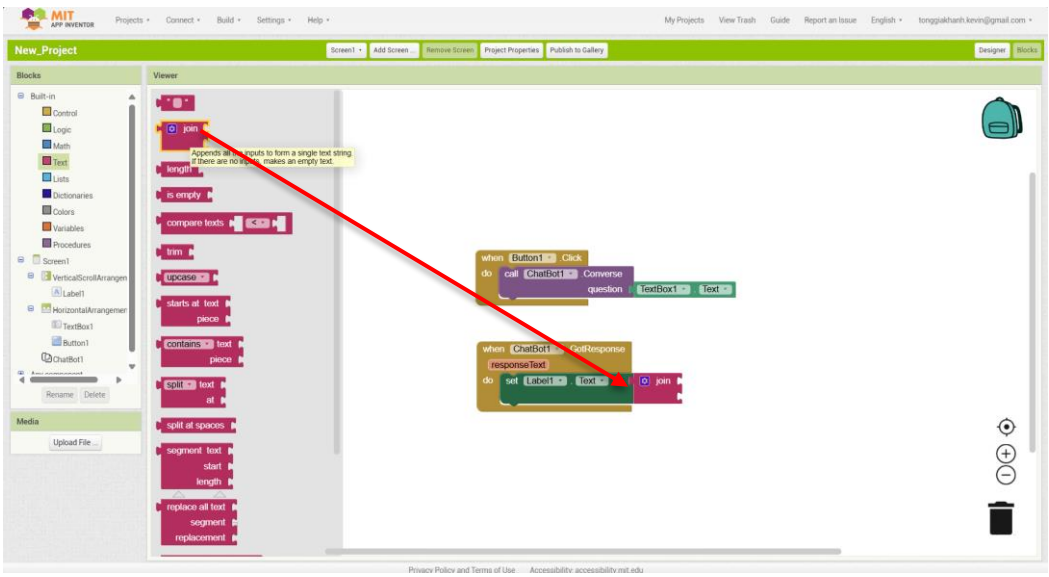

MIT APP INVENTOR
New_Project
Blocks
Viewer
when Button1 Click
do
when Button1 GotFocus
when Button1 LongClick
when Button1 LostFocus
when Button1 TouchDown
when Button1 TouchUp
Button1 BackgroundColor
set Button1 BackgroundColor to
Button1 Enabled
set Button1 Enabled to
Button1 FontBold
set Button1 FontBold to
Button1 FontItalic
when Button1 Click
do

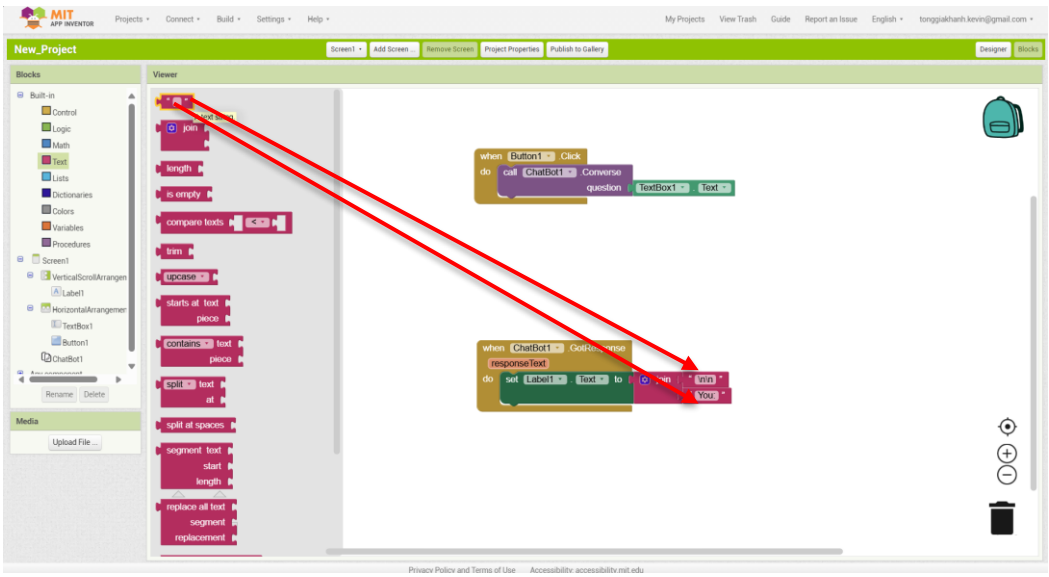

MIT APP INVENTOR
New_Project
Blocks
Viewer
when ChatBot1 ErrorOccurred
responseCode responseText
do
when ChatBot1 GotResponse
responseText
do
call ChatBot1 Converse
question
call ChatBot1 ConverseWithImage
question
source
call ChatBot1 ResetConversation
ChatBot1 ApiKey
set ChatBot1 ApiKey to
ChatBot1 Model
set ChatBot1 Model to chatgpt-gpt-4o-mini
ChatBot1 Provider
set ChatBot1 Provider to chatgpt
ChatBot1 System
set ChatBot1 System to
when Button1 Click
do call ChatBot1 Converse
question

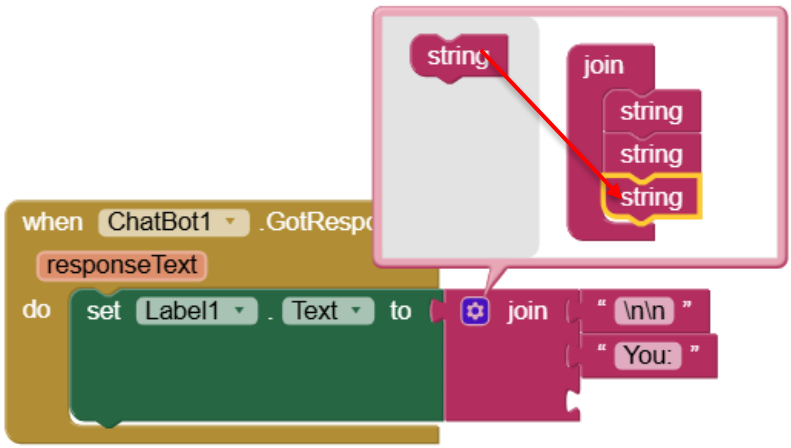

MIT APP INVENTOR
New_Project
Blocks
Viewer
TextBox1 FontSize
set TextBox1 FontSize to
TextBox1 Height
set TextBox1 Height to
set TextBox1 HeightPercent to
TextBox1 Hint
set TextBox1 Hint to
TextBox1 HintColor
set TextBox1 HintColor to
TextBox1 MultiLine
set TextBox1 MultiLine to
TextBox1 NumbersOnly
set TextBox1 NumbersOnly to
TextBox1 ReadOnly
set TextBox1 ReadOnly to
TextBox1 Text
set TextBox1 Text to
when Button1 Click
do call ChatBot1 Converse
question TextBox1 Text

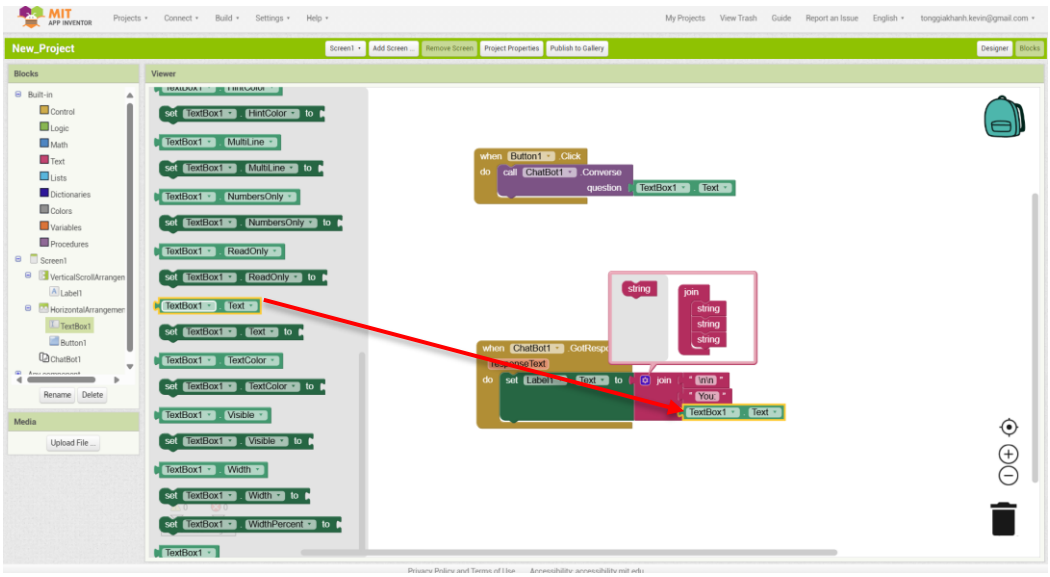

when Button1 .Click
do call ChatBot1 .Converse
question TextBox1 . Text

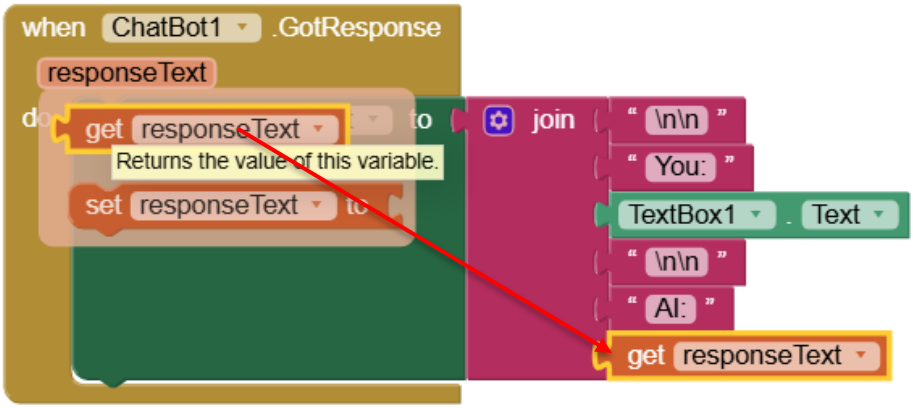

MIT APP INVENTOR
New_Project
Blocks
Viewer
when Button1 .Click
do call ChatBot1 .Converse
question TextBox1 . Text
when ChatBot1 .GotResponse
responseText
set Label1 . Text to

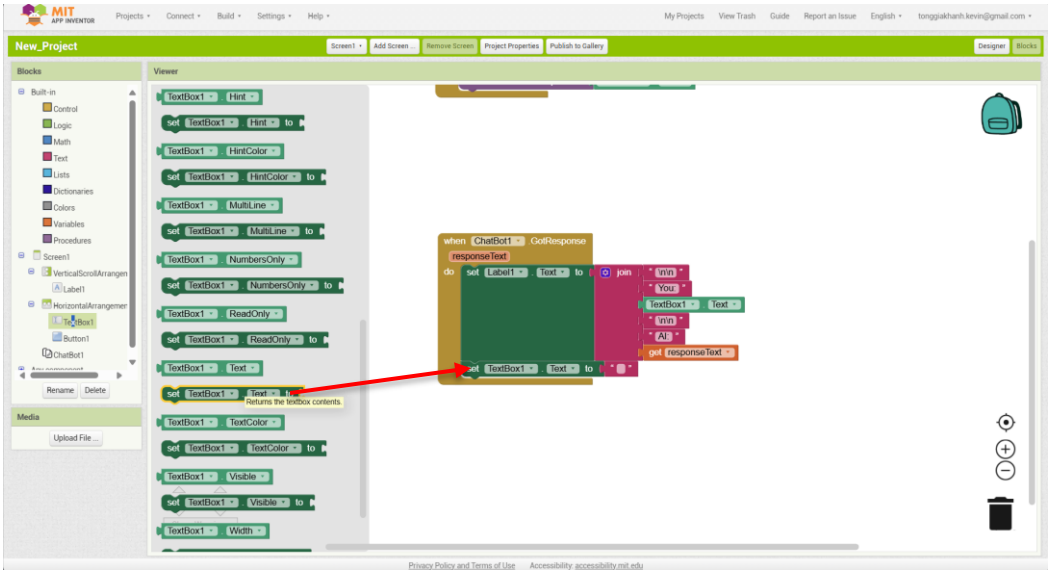

MIT APP INVENTOR
New_Project
Blocks
Viewer
join
when Button1 .Click
do call ChatBot1 .Converse
question TextBox1 . Text
when ChatBot1 .GotResponse
responseText
do set Label1 . Text to join


MIT APP INVENTOR
New_Project
Blocks
Viewer
when Button1 Click
do call ChatBot1 Converse
question TextBox1 Text
when ChatBot1 GotResponse
responseText
do set Label1 Text to join "\n\n" "You:"

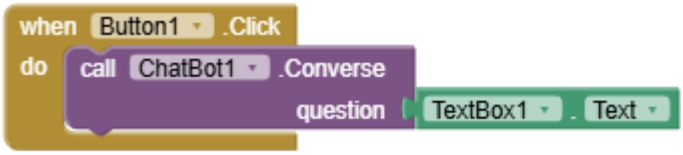

string
join
string
string
string
when ChatBot1 .GotResp
responseText
do set Label1 . Text to join "\n\n" "You:"

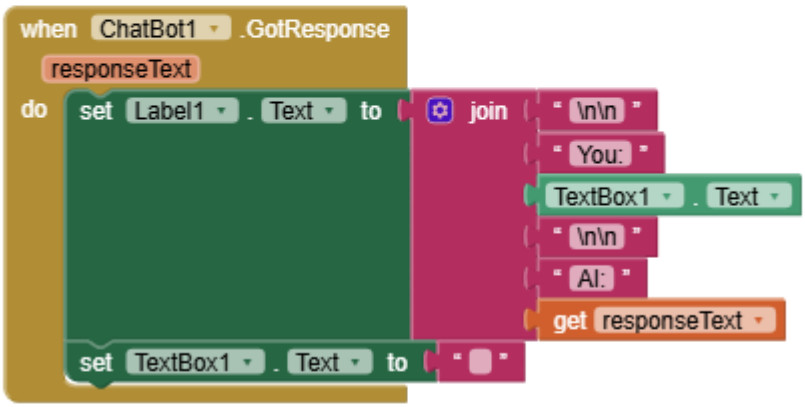

MIT APP INVENTOR
New_Project
Blocks
Viewer
when Button1 Click
do call ChatBot1 Converse
question TextBox1 Text
when ChatBot1 GotResp
responseText
do set Label1 Text to join "\n\n" "You:"
TextBox1 Text



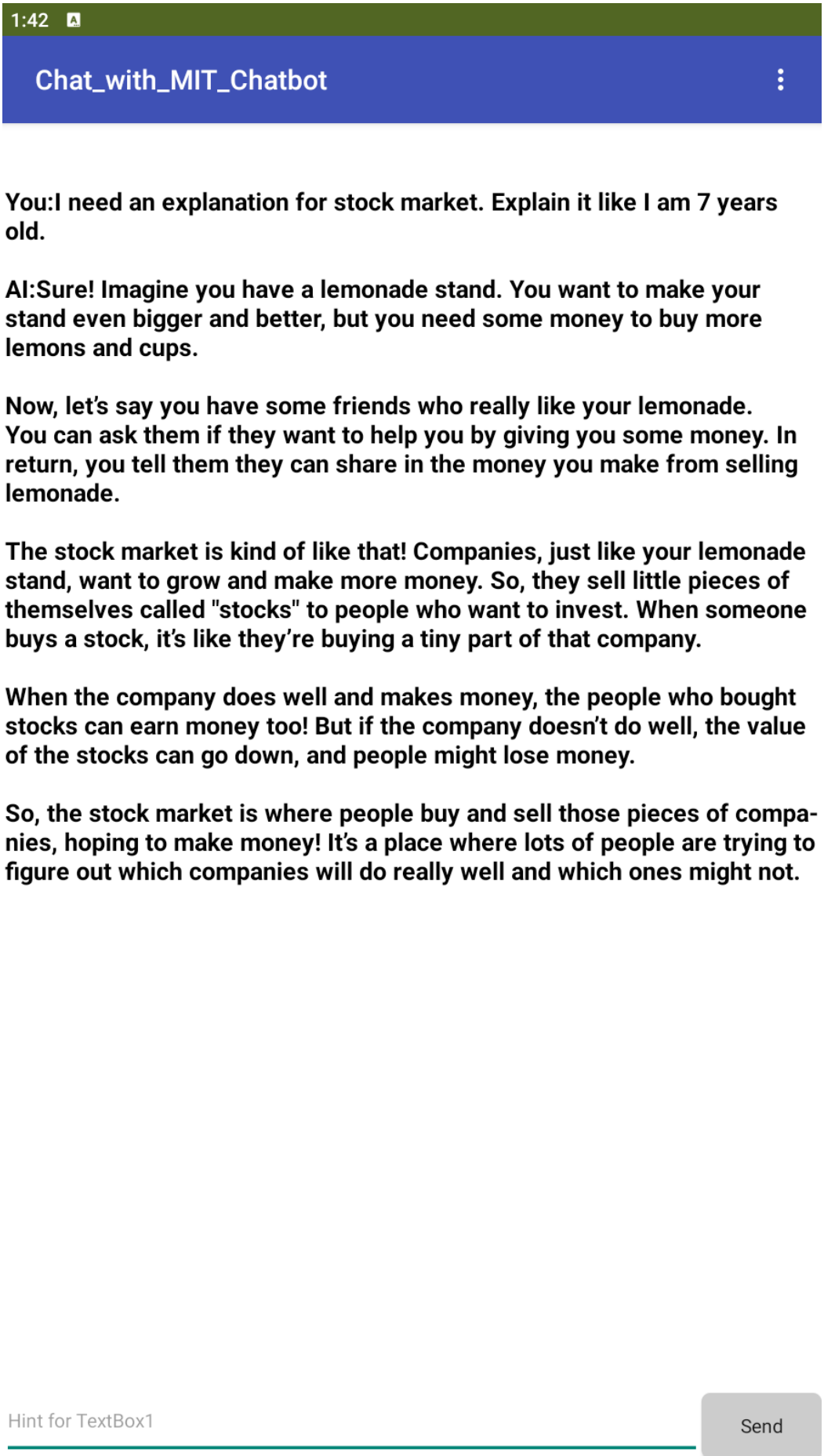


Completed CODE Blocks:


when Button1 .Click
do call ChatBot1 .Converse
question TextBox1 . Text


when ChatBot1 .GotResponse
responseText
do set Label1 . Text to join
" \n\n "
" You: "
TextBox1 . Text
" \n\n "
" AI: "
get responseText
set TextBox1 . Text to " "

DEMO:



MCQ:
What is the primary role of the ChatBot1.Converse block?

A) To send a predefined message.
B) To process the user's input and return a response.
C) To clear the text box after a response.
D) To display the chatbot's name.
(Correct Answer: B)

1. Which component in the front end is responsible for displaying the chat history?

   A) TextBox1
   B) Label1
   C) ChatBot1
   D) Button1
   (Correct Answer: B)

2. Why is the TextBox1.Text cleared in the GotResponse block?

   A) To allow the user to type a new question.
   B) To save memory.
   C) To display the chatbot's answer.
   D) To end the conversation.
   (Correct Answer: A)

3. What will happen if the Label1.Text block is not set to include the user's input in the GotResponse block?

   A) The app will crash.
   B) The chatbot's response will not display.
   C) The user's input will not appear in the conversation history.
   D) Nothing will happen.
   (Correct Answer: C)

Application Qn:

Question 1: Adding Custom Responses

Challenge:

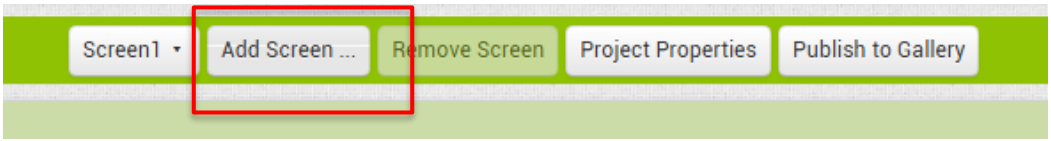
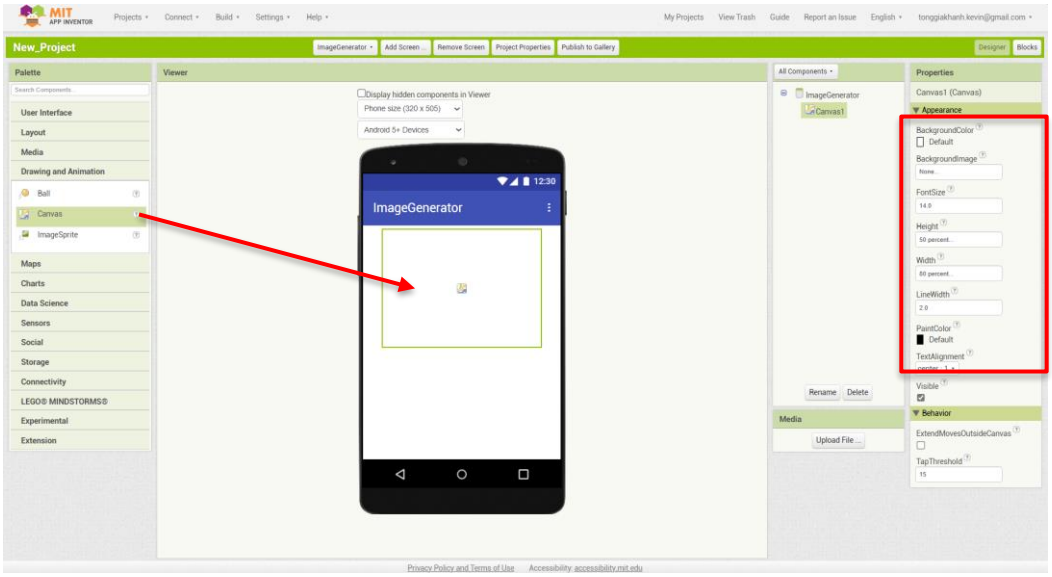
- The chatbot currently provides generic responses to user inputs. Suppose you want the chatbot to recognize and provide custom responses for specific keywords. For example, when the user types "weather," the chatbot should respond with "The weather is sunny today."

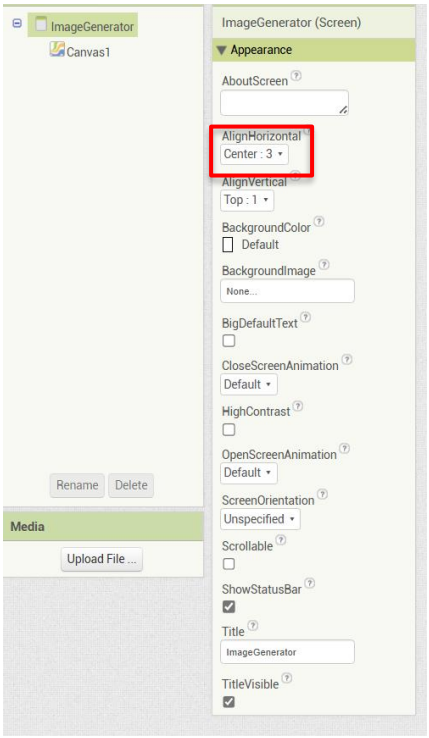
Answer:

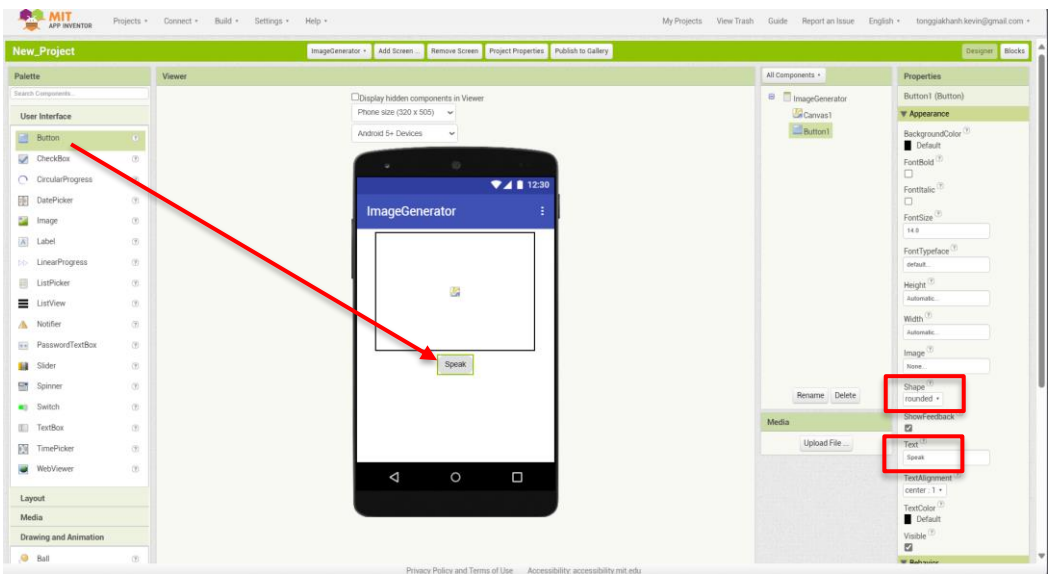
- Modify the GotResponse block.
- Before setting the Label1.Text, add a conditional check using the if block:

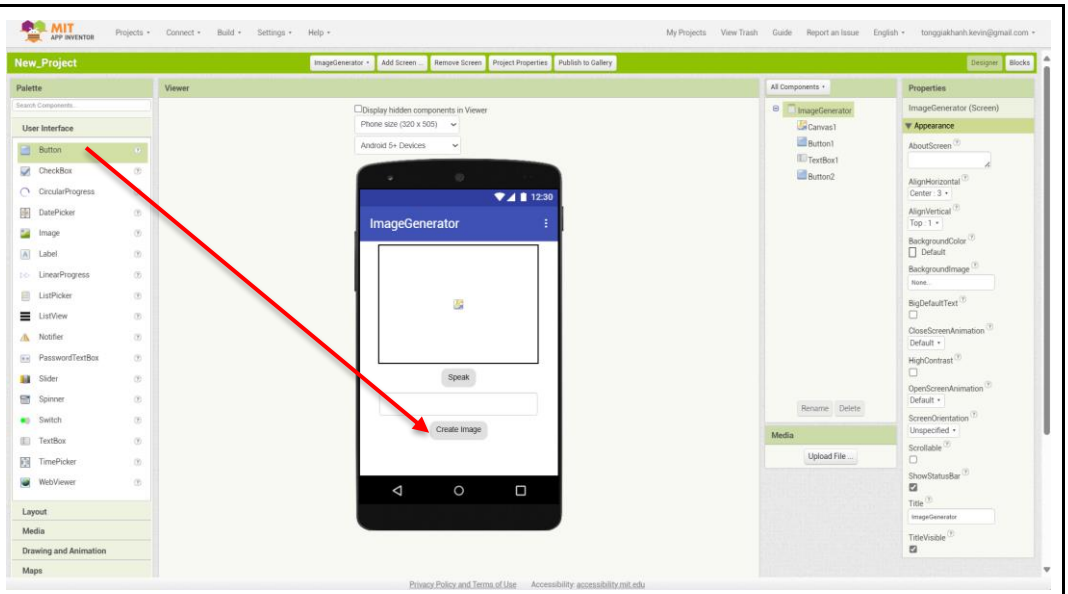
	• If TextBox1.Text contains the word "weather," set Label1.Text to "The weather is sunny today." • Use an else statement to call the default chatbot response for all other inputs. • This customization enhances the app's interactivity by adding personalized responses. ***Question 2: Formatting Chat Display*** Challenge: • Currently, the chatbot's responses and user input appear together in the Label1 component. How can you modify the app to separate the user messages and chatbot responses into two different labels (e.g., UserLabel for user input and BotLabel for chatbot responses)? Answer: • Add a new label component (BotLabel) below the existing UserLabel. • In the Button1.Click block, set UserLabel.Text to display the user's input. • In the GotResponse block, set BotLabel.Text to display the chatbot's response. • Adjust the design layout using vertical or horizontal arrangements to align the labels for readability. • This setup creates a cleaner and more professional chat interface.

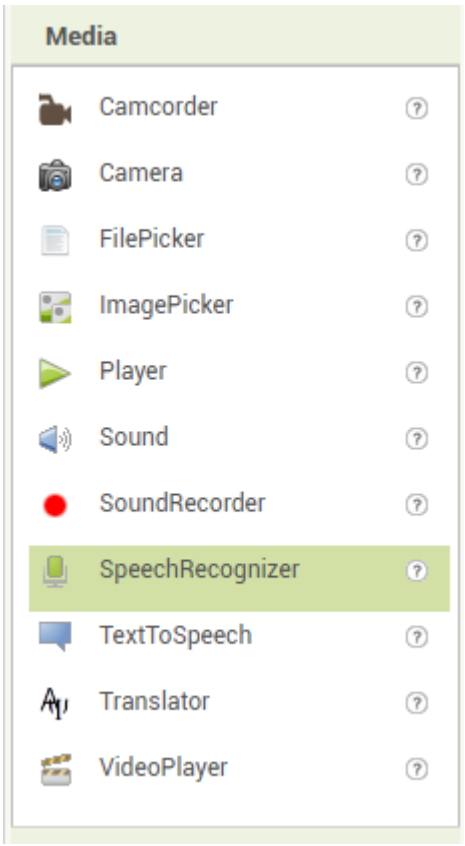
Project 2 – AI Image Generator (Kevin) (hrs)	
Learning Outcomes (5mins)	**Learning Outcome 1: Learn to Integrate AI-Powered Image Generation into App Development:** ● Students will understand how to incorporate AI image generation functionality into an app, linking user input commands to generated visual outputs. **Learning Outcome 2: Develop Skills in Managing Media Components and Display Mechanisms:** ● Students will learn to handle image components and display results dynamically on a canvas, ensuring seamless integration with the user interface.

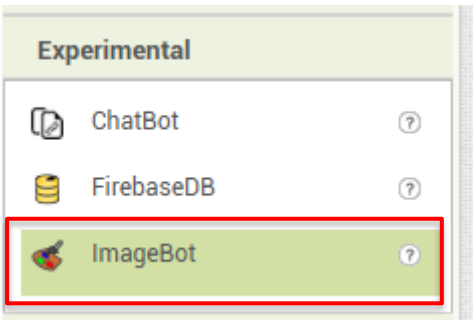
	Learning Outcome 3: Understand and Utilize Key Functions and Variables in App Development: Utilize Key Programming Constructs for Data and Media Handling ● Students will gain experience in using functions, variables, and commands to process user inputs, retrieve AI-generated images, and display them efficiently in the app.
Session No.: Session Title (hrs)	**Project Brief** ● This project teaches students how to build an AI Image Generator app using MIT App Inventor. Students will develop both the interface and backend functionality, enabling users to input commands to generate desired images. The app integrates AI-generated results with image components, displaying them dynamically on a canvas for a seamless user experience. **Task:**  

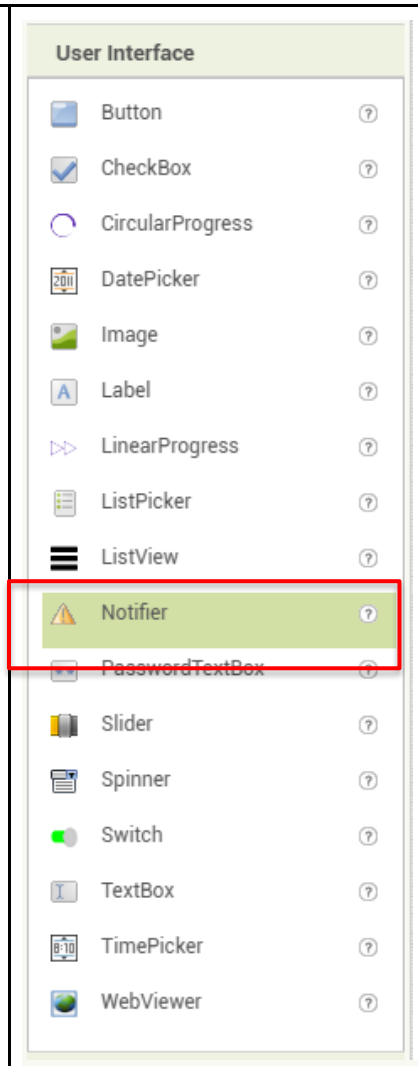

ImageGenerator
Canvas1
Rename
Delete
Media
Upload File ...
ImageGenerator (Screen)
Appearance
AboutScreen
AlignHorizontal
Center : 3
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
ImageGenerator
TitleVisible

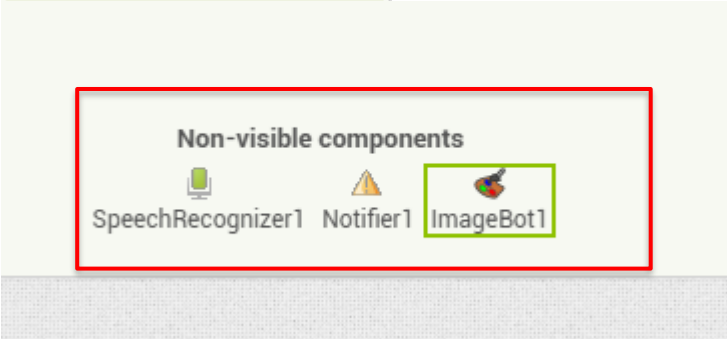

New_Project
Palette
Viewer
ImageGenerator
Speak
Properties
Button1 (Button)
Appearance
Shape
rounded
Text
Speak
Media
Upload File ...
Rename
Delete

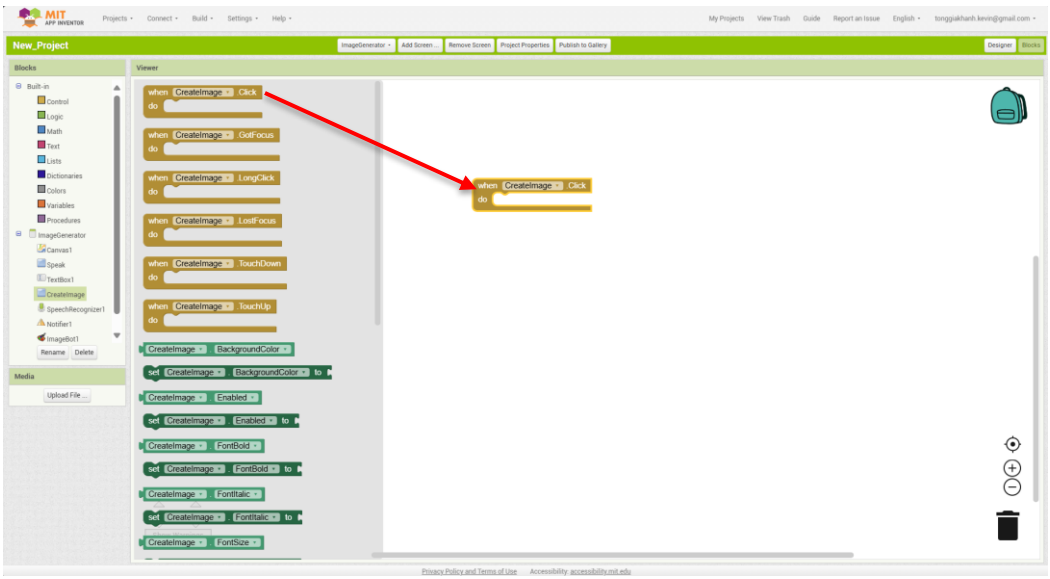

MIT APP INVENTOR
New_Project
Palette
User Interface
Button
CheckBox
CircularProgress
DatePicker
Image
Label
LinearProgress
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Viewer
Display hidden components in Viewer
Phone size (320 x 505)
Android 5+ Devices
ImageGenerator
Speak
Create Image
All Components
ImageGenerator
Canvas1
Button1
TextBox1
Button2
Rename
Delete
Media
Upload File ...
Properties
ImageGenerator (Screen)
Appearance
AboutScreen
AlignHorizontal
AlignVertical
BackgroundColor
BackgroundImage
BigDefaultText
CloseScreenAnimation
HighContrast
OpenScreenAnimation
ScreenOrientation
Scrollable
ShowStatusBar
Title
TitleVisible

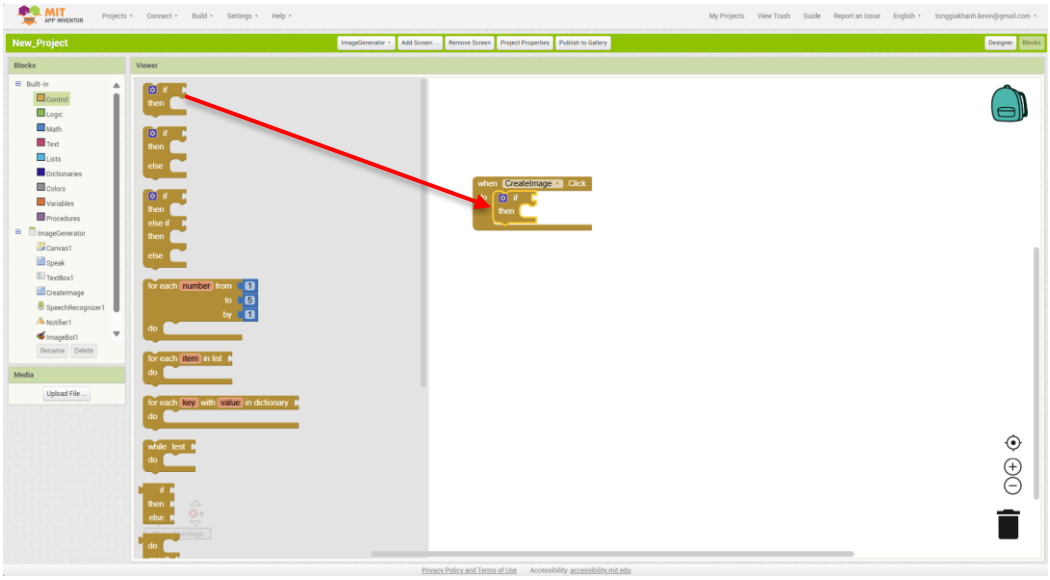

Media
Camcorder
Camera
FilePicker
ImagePicker
Player
Sound
SoundRecorder
SpeechRecognizer
TextToSpeech
Translator
VideoPlayer

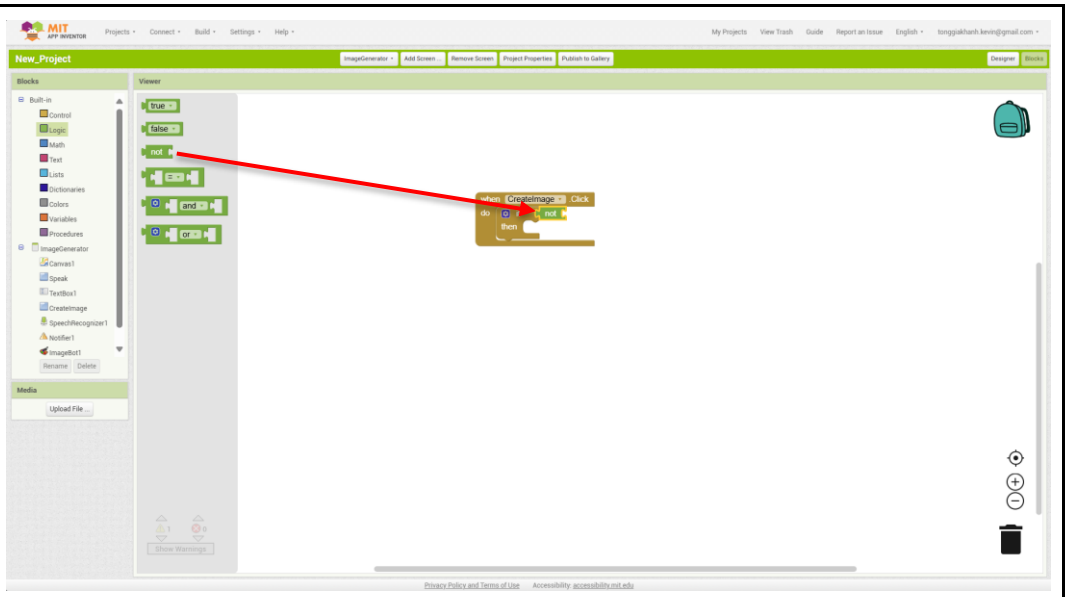

Experimental
ChatBot
FirebaseDB
ImageBot

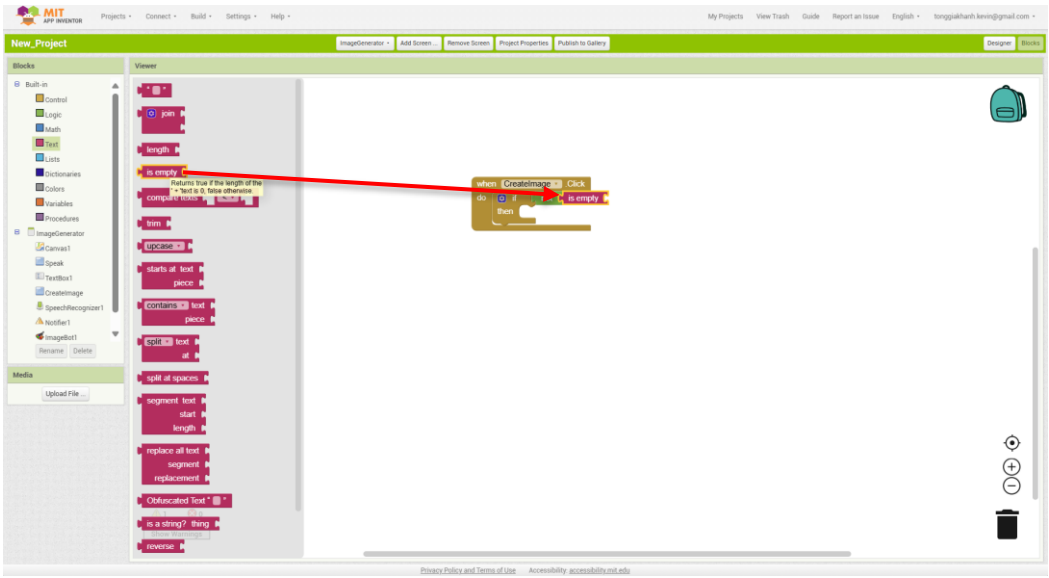

User Interface
Button
CheckBox
CircularProgress
DatePicker
Image
Label
LinearProgress
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer

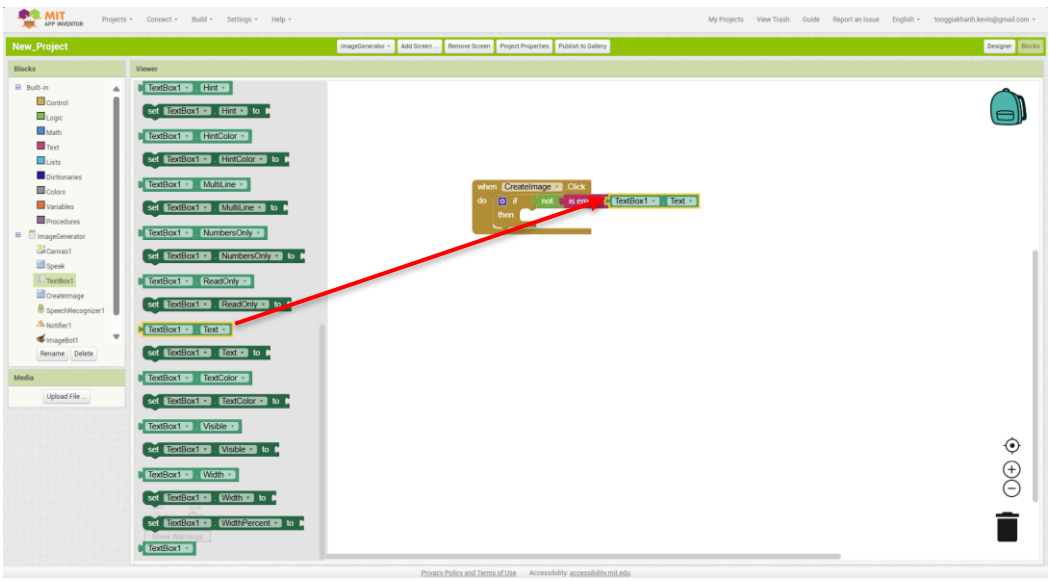

Non-visible components
SpeechRecognizer1
Notifier1
ImageBot1

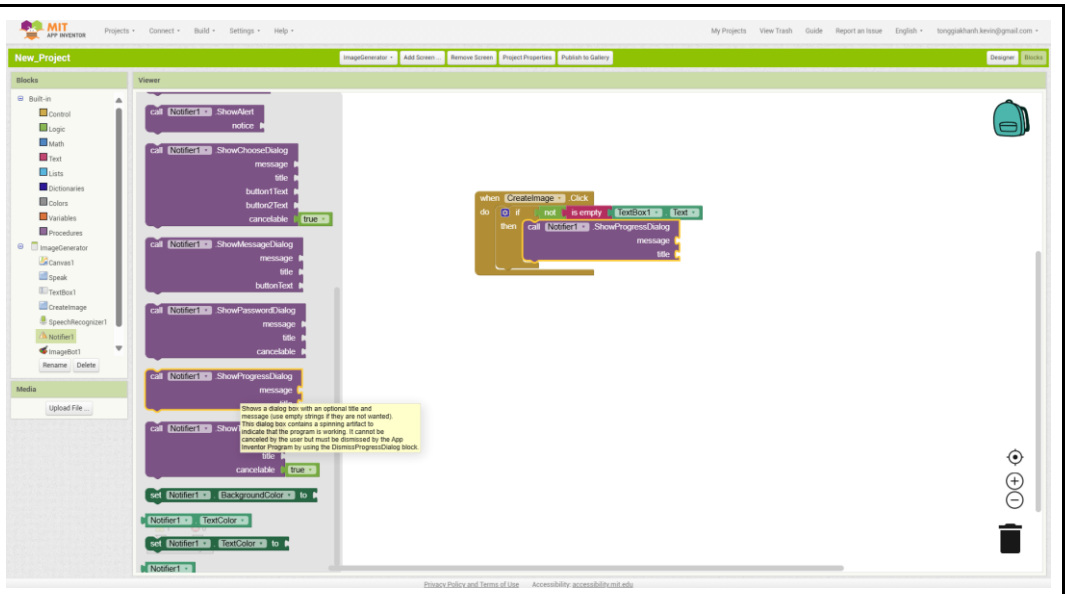

MIT APP INVENTOR
New_Project
Blocks
Viewer
when CreateImage Click
do
when CreateImage GotFocus
when CreateImage LongClick
when CreateImage LostFocus
when CreateImage TouchDown
when CreateImage TouchUp
CreateImage BackgroundColor
set CreateImage BackgroundColor to
CreateImage Enabled
set CreateImage Enabled to
CreateImage FontBold
set CreateImage FontBold to
CreateImage FontItalic
set CreateImage FontItalic to
CreateImage FontSize

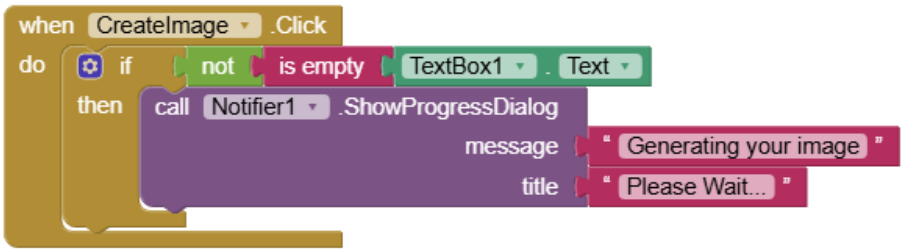

MIT APP INVENTOR
New_Project
Blocks
Viewer
if then
if then else
if then else if then else
for each number from 1 to 5 by 1
for each item in list
for each key with value in dictionary
while test
when CreateImage Click
if then

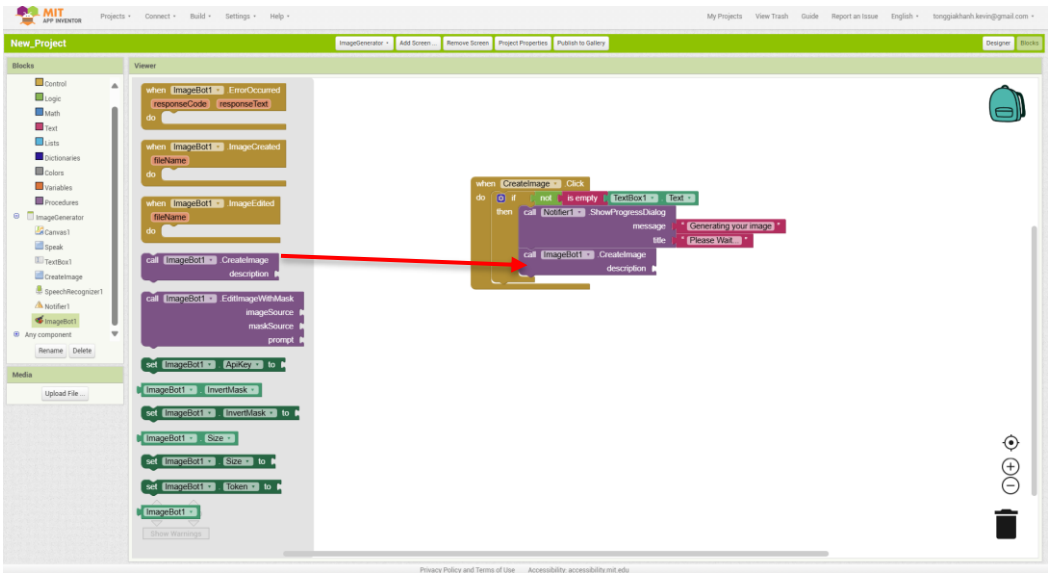

New_Project
Blocks
Viewer
true
false
not
and
or
when CreateImage Click
do if then not

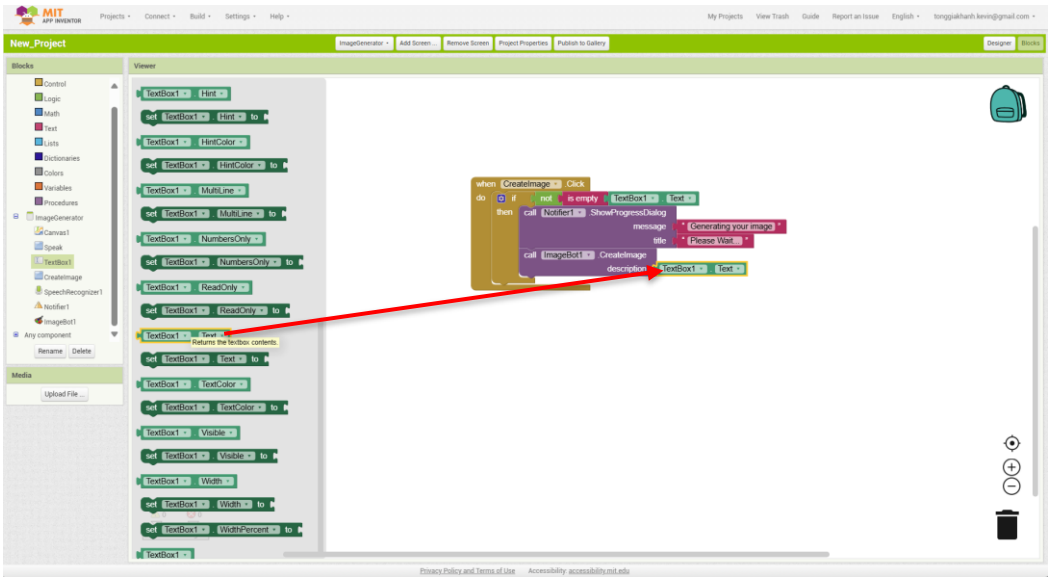

New_Project
Blocks
Viewer
join
length
is empty
Returns true if the length of the text is 0, false otherwise.
trim
upcase
starts at text piece
contains text piece
split text at
split at spaces
segment text start length
replace all text segment replacement
Obfuscated Text
is a string? thing
reverse
when CreateImage Click
do if then is empty

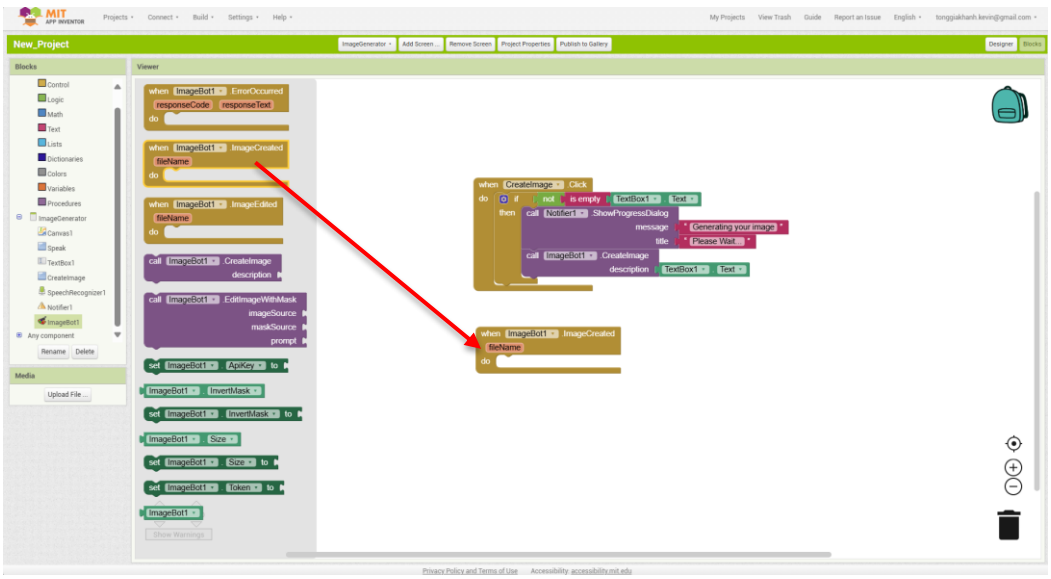

New_Project
Blocks
Viewer
TextBox1 Hint
set TextBox1 Hint to
TextBox1 HintColor
set TextBox1 HintColor to
TextBox1 MultiLine
set TextBox1 MultiLine to
TextBox1 NumbersOnly
set TextBox1 NumbersOnly to
TextBox1 ReadOnly
set TextBox1 ReadOnly to
TextBox1 Text
set TextBox1 Text to
TextBox1 TextColor
set TextBox1 TextColor to
TextBox1 Visible
set TextBox1 Visible to
TextBox1 Width
set TextBox1 Width to
set TextBox1 WidthPercent to
TextBox1
when CreateImage Click
do if not is empty TextBox1 Text then

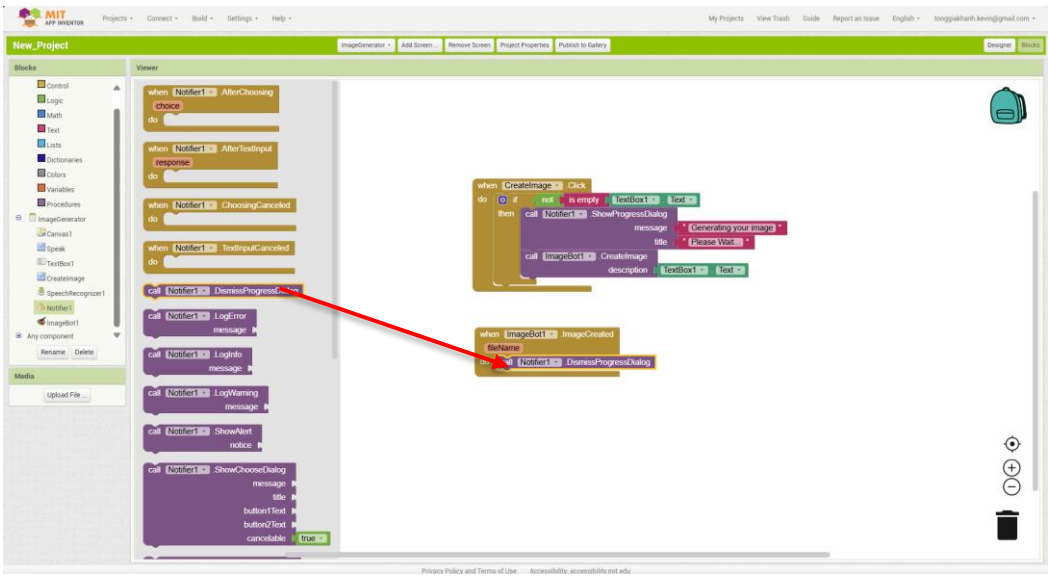


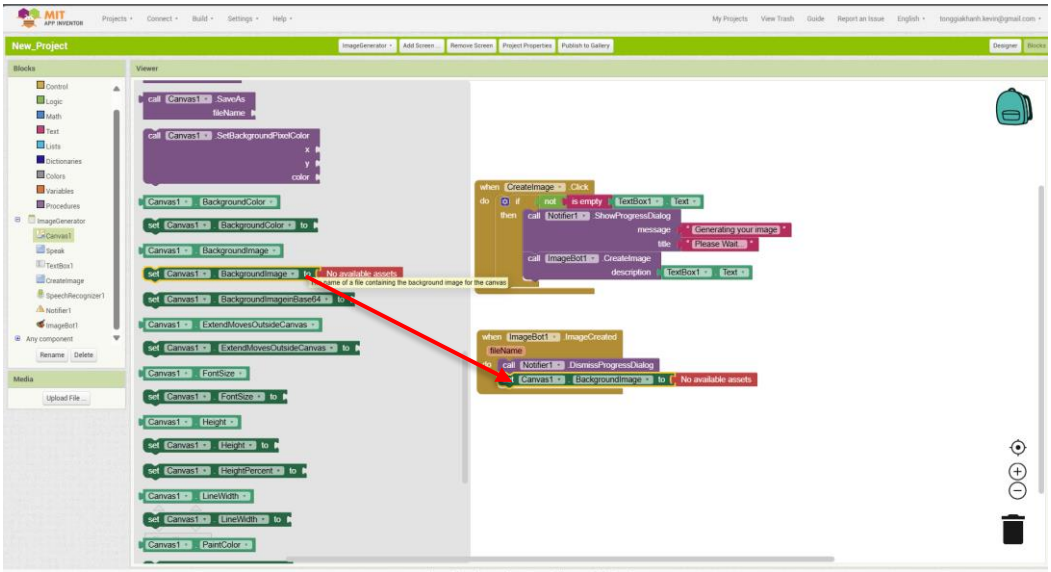

when CreateImage .Click
do if not is empty TextBox1 . Text
then call Notifier1 .ShowProgressDialog
message " Generating your image "
title " Please Wait... "

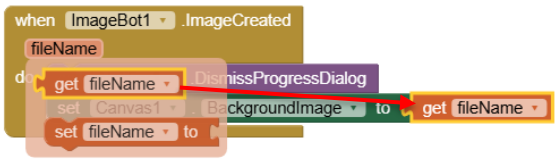


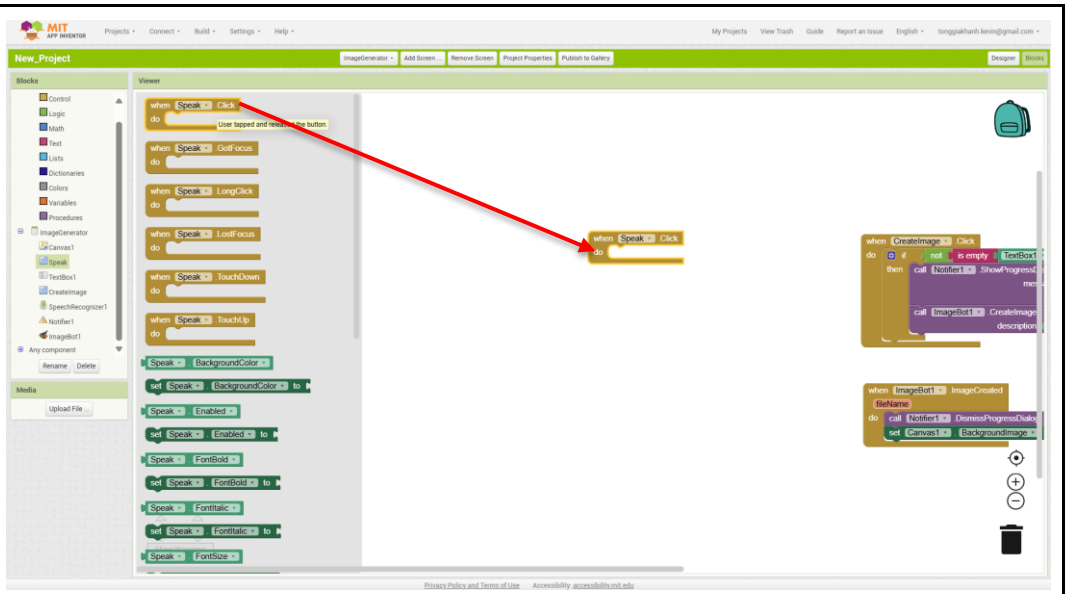

New_Project
when CreateImage .Click
do if not is empty TextBox1 Text
then call Notifier1 ShowProgressDialog
message "Generating your image"
title "Please Wait"
call ImageBot1 CreateImage
description TextBox1 Text

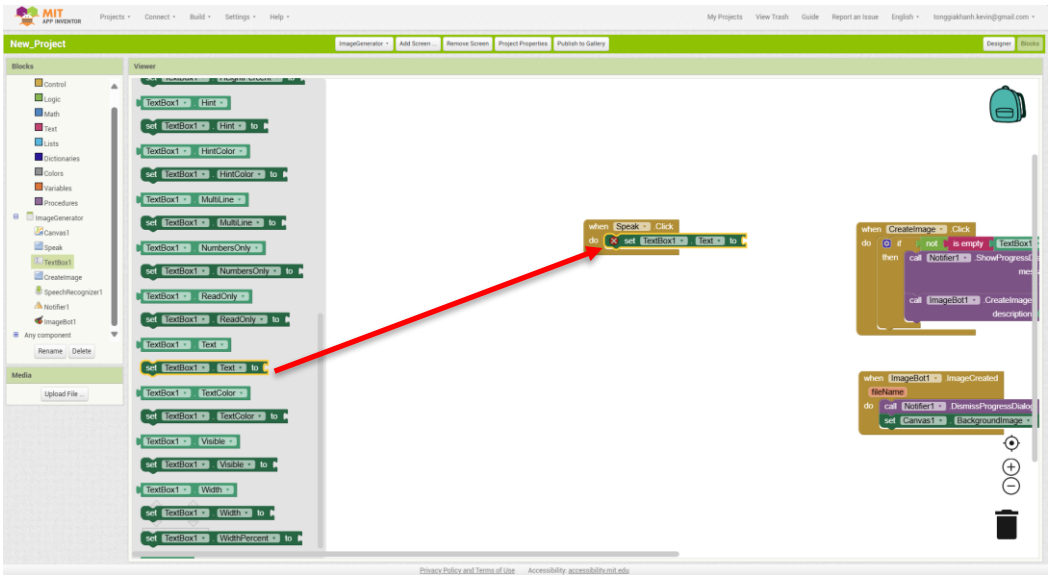

New_Project
when ImageBot1 ErrorOccurred
when ImageBot1 ImageCreated
fileName
when ImageBot1 ImageEdited
call ImageBot1 CreateImage
call ImageBot1 EditImageWithMask
when CreateImage .Click
when ImageBot1 ImageCreated

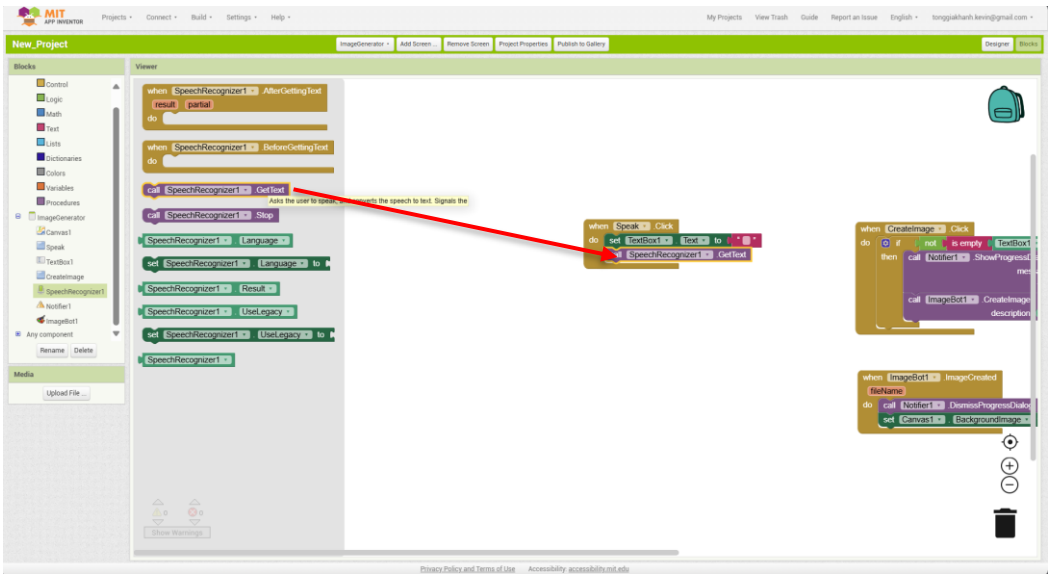

New_Project
when Notifier1 AfterChoosing
when Notifier1 AfterTextInput
when Notifier1 ChoosingCanceled
when Notifier1 TextInputCanceled
call Notifier1 LogError
call Notifier1 LogInfo
call Notifier1 LogWarning
call Notifier1 ShowAlert
call Notifier1 ShowChooseDialog
when CreateImage .Click
when ImageBot1 ImageCreated
call Notifier1 DismissProgressDialog

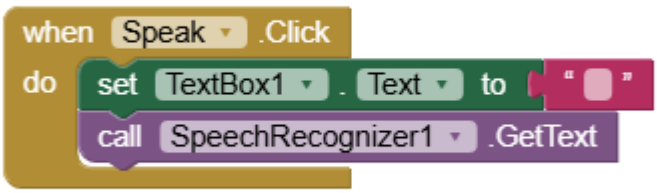


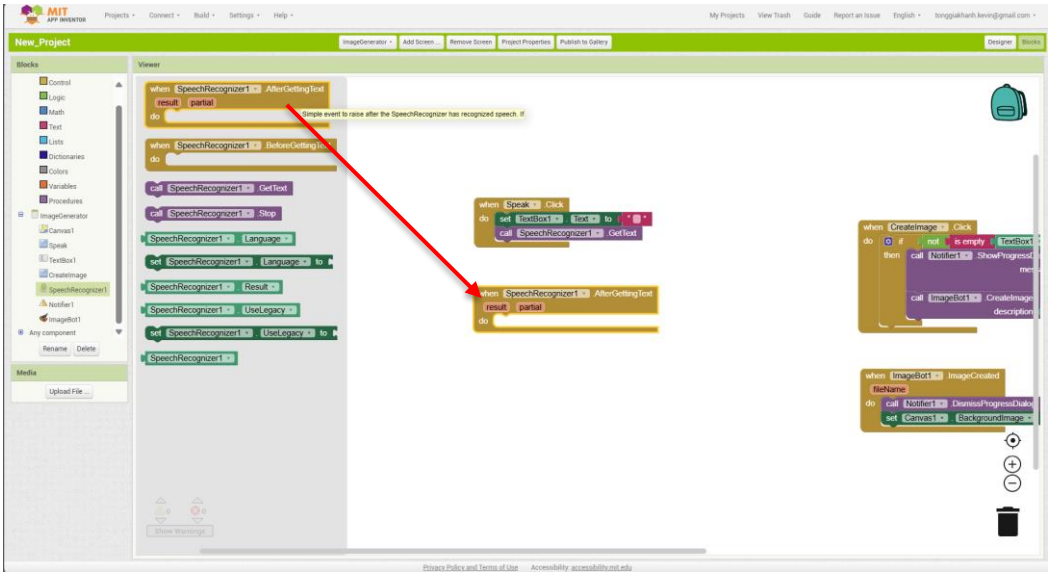

when ImageBot1 .ImageCreated
fileName
get fileName
DismissProgressDialog
set Canvas1 . BackgroundImage to get fileName
set fileName to

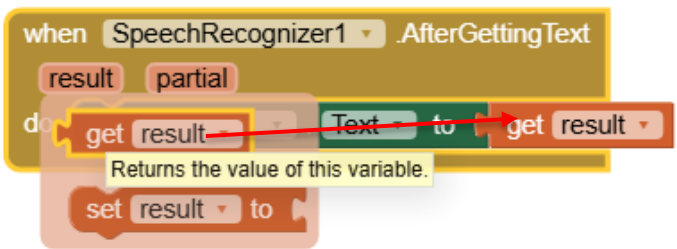

New_Project
when Speak Click
when Speak GotFocus
when Speak LongClick
when Speak LostFocus
when Speak TouchDown
when Speak TouchUp
when CreateImage Click
when ImageBot1 ImageCreated

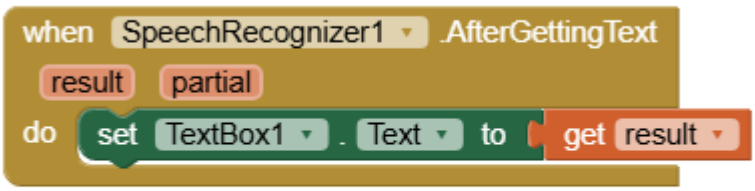

New_Project
when Speak Click
when CreateImage Click
when ImageBot1 ImageCreated


New_Project
when SpeechRecognizer1 AfterGettingText
when SpeechRecognizer1 BeforeGettingText
call SpeechRecognizer1 GetText
when Speak Click
when CreateImage Click
when ImageBot1 ImageCreated

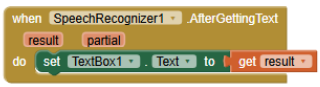

when Speak .Click
do set TextBox1 . Text to " "
call SpeechRecognizer1 .GetText

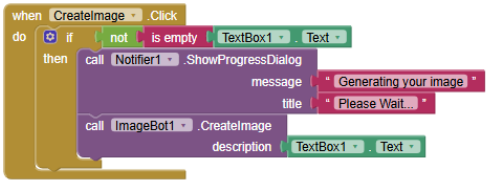


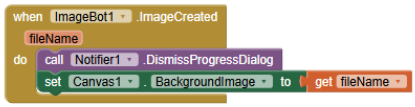

when SpeechRecognizer1 .AfterGettingText
result partial
do get result Text to get result
Returns the value of this variable.
set result to


when SpeechRecognizer1 .AfterGettingText
result partial
do set TextBox1 . Text to get result

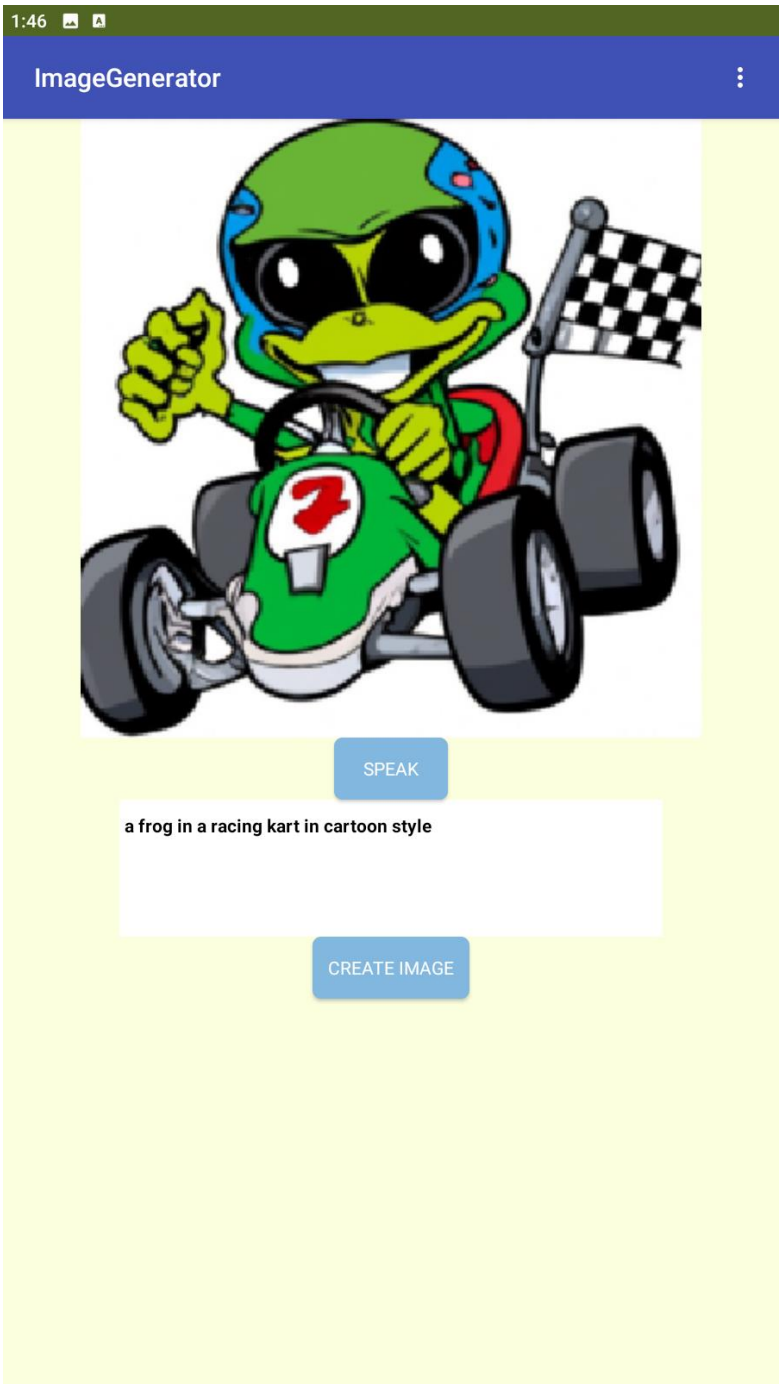
Completed Code blocks:







DEMO:



MCQs:

Question 1:

What is the purpose of the Notifier component in the Image Generator app?

A. To generate images.
B. To display progress or alert messages to the user.
C. To manage speech-to-text conversion.
D. To act as a backend server for image generation.

Correct Answer: B

Question 2:

What happens when the SpeakButton is clicked?

A. The text in TextBox1 is converted to speech.
B. The app sends the text to ImageBot1 for image generation.
C. The SpeechRecognizer component captures speech and converts it to text for TextBox1.
D. The app displays an error message if no input is detected.

Correct Answer: C

Question 3:

Which block sets the generated image as the background of the canvas?

A. Notifier1.ShowProgressDialog.
B. SpeechRecognizer1.AfterGettingText.
C. ImageBot1.ImageCreated.
D. CreateImage.Click.

Correct Answer: C

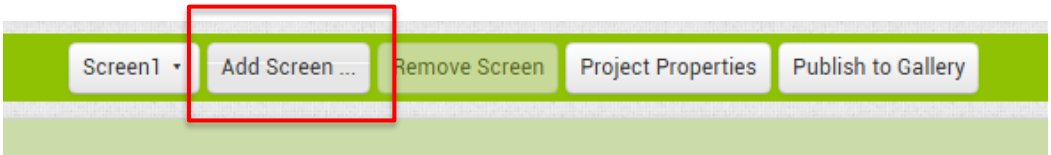
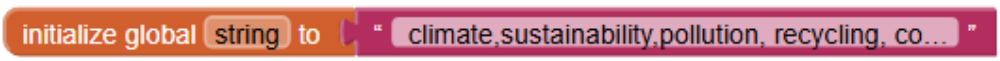
Question 4:

What condition is checked before initiating the image generation process?

A. If the microphone is working properly.
B. If the user clicks the SpeakButton.
C. If TextBox1.Text is not empty.
D. If the canvas has an existing background.

Correct Answer: C

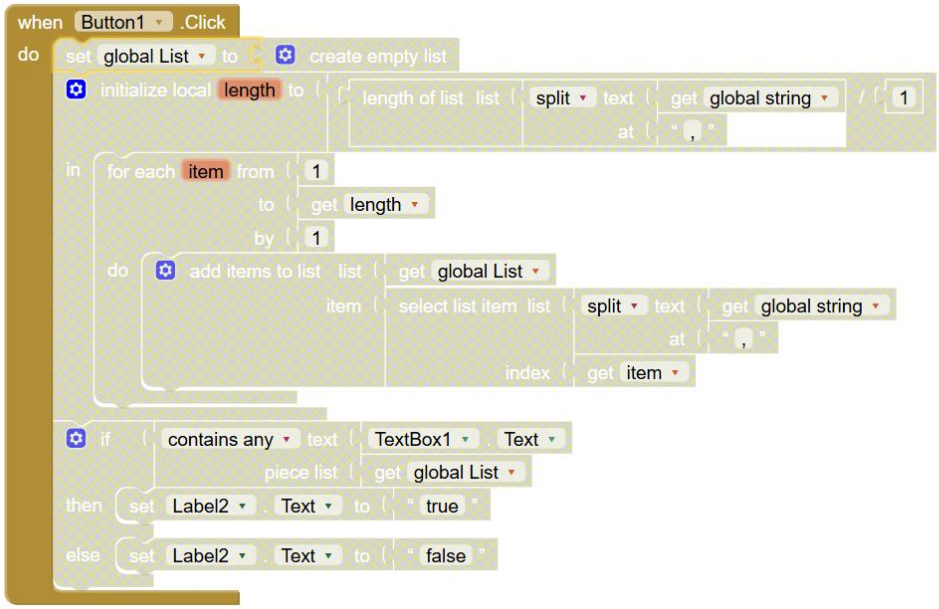
	Application Qn: **Challenge:** Imagine a user clicks the "Create Image" button without entering text into the `TextBox1`. The app currently checks if `TextBox1` is empty before proceeding. However, the user claims they didn't notice the prompt. How would you modify the app to provide a clearer error message and ensure a better user experience? **Answer:** To address the challenge: 1. **Update Logic:** Modify the blocks inside the `CreateImage.Click` event to include a clearer error message if `TextBox1` is empty. o Instead of silently not proceeding, add a `Notifier1.ShowAlert` block to alert the user with a message like *"Please enter a description before generating an image."* 2. **Implementation:** Update the `if not is empty(TextBox1.Text)` condition by adding an `else` block that contains: o `call Notifier1.ShowAlert` o Set `message` to: *"Error: You must input text to generate an image!"* **Enhanced User Experience:** By adding this explicit feedback, the app informs users why the operation failed, ensuring they understand what action to take next. This makes the app more intuitive and user-friendly. Would you like a visual explanation of the updated blocks? **WAY TO IMPROVE, how can students use the imagebot. Image edited to add an extra function?**

Project 3 – Filtered input AI (Kevin) (hrs)	
Learning Outcomes (5mins)	**Learning Outcome 1 Understanding Topic-Based Filtering:** ● Students will learn how to implement a filtering system that validates user input against a predefined set of keywords to restrict chatbot responses to relevant topics. **Learning Outcome 2 Dynamic List Management and String Manipulation** ● Students will develop skills in creating and managing lists, splitting strings, and comparing input data to predefined values using logical and iterative blocks. **Learning Outcome 3 Logical Flow and Decision Making in App Development:** ● Students will understand how to use conditionals to control app behavior, ensuring appropriate responses (true/false) based on the validity of user input.
Session No.: Session Title (hrs)	**Project Brief** ● This project aims to enhance two AI-powered apps—a chatbot and an image generator—by integrating a topic-filtering system tailored for sustainability and environmental themes. The filtering mechanism ensures that only relevant inputs, such as questions or commands related to climate change, recycling, and renewable energy, are processed. By refining the apps to focus solely on these topics, the project elevates their precision and impact, transforming them into purpose-driven tools for education and advocacy. This ensures the final product aligns with the Hackathon's theme of sustainability, addressing key environmental challenges with relevance and effectiveness. **Task:** **1.**  **2.** 

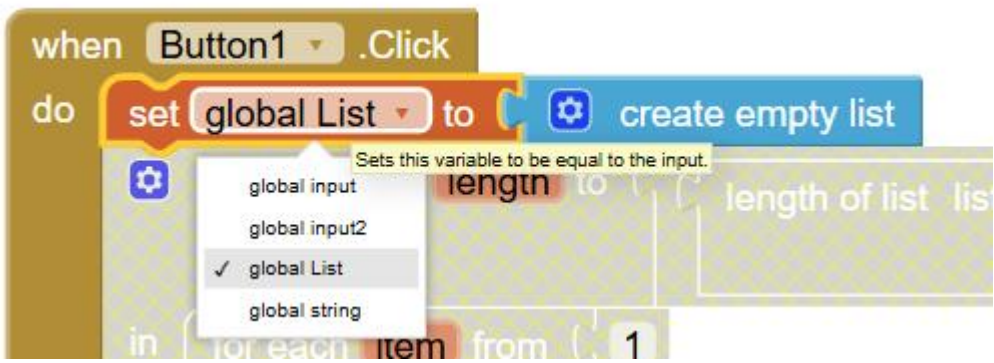
3.


initialize global List to
create empty list

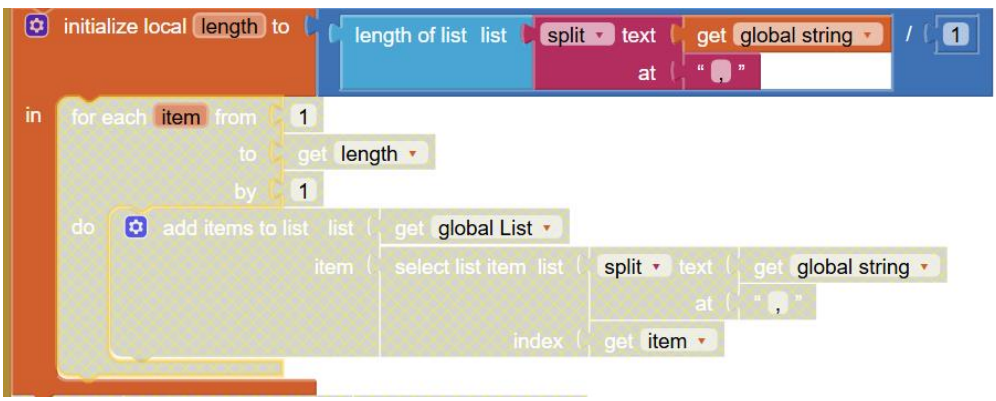
4.


when Button1 .Click
do set global List to create empty list
initialize local length to length of list list split text get global string / 1
at ","
in for each item from 1
to get length
by 1
do add items to list list get global List
item select list item list split text get global string
at ","
index get item
if contains any text TextBox1 . Text
piece list get global List
then set Label2 . Text to "true"
else set Label2 . Text to "false"

5.


when Button1 .Click
do set global List to create empty list
Sets this variable to be equal to the input.
global input
global input2
global List
global string
length of list lis
in for each item from 1

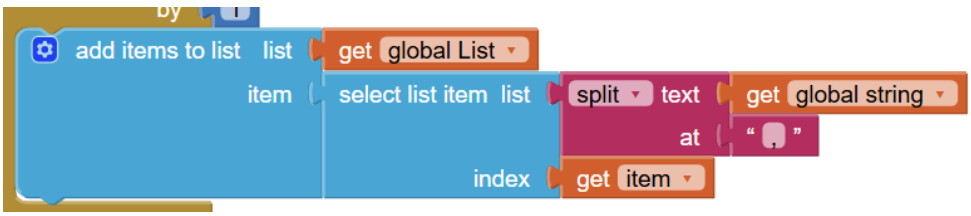
6.


initialize local length to length of list list split text get global string / 1
at ","
in for each item from 1
to get length
by 1
do add items to list list get global List
item select list item list split text get global string
at ","
index get item

7.

8.

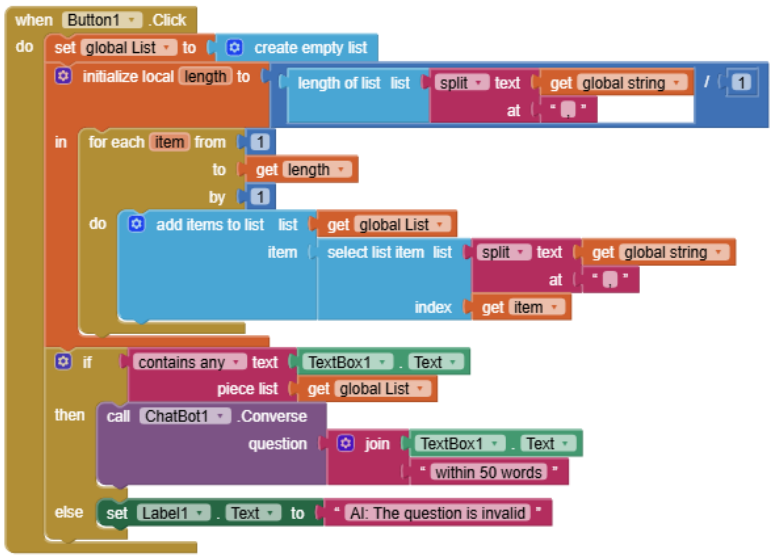
9.



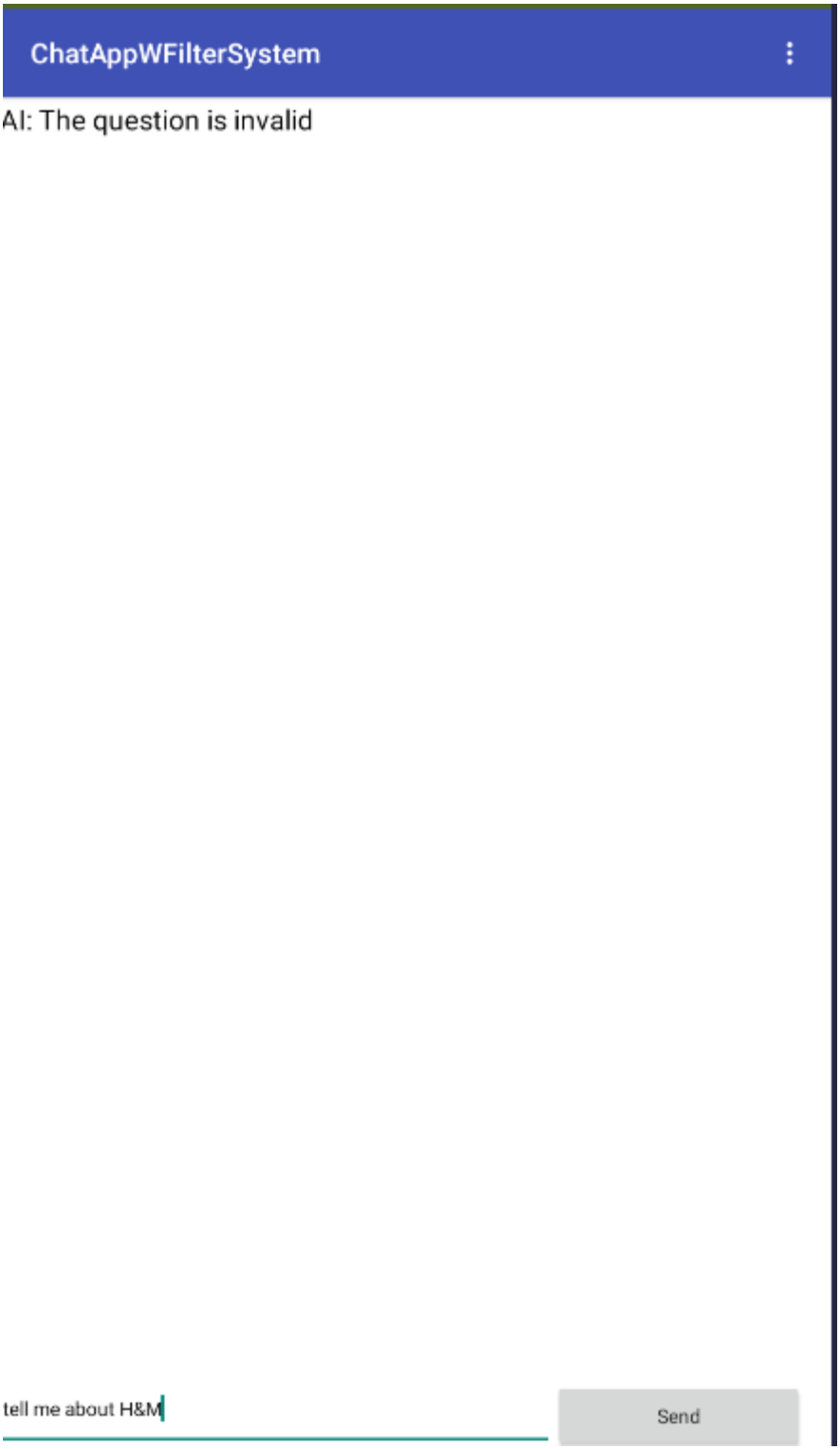
10.

TO CONNECT PROJECT 3 to PROJECT 1 or PROJECT 2.
->Replace Green box blocks (then & else) with the according code.

Completed code blocks:



DEMO:



MCQs:

1. What is the primary purpose of the topic-filtering system in this project?

A. To increase the chatbot's response speed
B. To ensure inputs align with sustainability and environmental topics
C. To allow the chatbot to handle more diverse questions
D. To reduce the complexity of the AI tools

Correct Answer: B

2. In the input filtering system, how are keywords compared to the

user's input?
A. By directly matching the input string with the predefined keywords
B. By splitting the input string into words and checking for overlaps with the keyword list
C. By using machine learning to predict valid topics
D. By ignoring all inputs and responding only with preset answers

Correct Answer: B

3. What happens when a user's input does not match any of the predefined sustainability keywords?
A. The system forwards the input to the chatbot regardless
B. The input is rejected, and the user is notified to ask a relevant question
C. The system modifies the input to make it relevant
D. The chatbot generates a random response

Correct Answer: B

4. Why is it important to limit the chatbot and image generator to specific topics for this project?
A. To make the apps simpler to use
B. To ensure the apps align with the Hackathon's sustainability theme and remain focused
C. To reduce the amount of data processed by the system
D. To improve the visual appeal of the app interface

Correct Answer: B

Application Qn:

Challenge:

Imagine you want to expand the topic-filtering system to allow for multi-language support. Users should be able to ask questions about sustainability in different languages (e.g., Spanish, French, or Mandarin). How would you modify the filtering mechanism to handle this while ensuring accuracy in identifying relevant topics? Outline your approach and describe how you would test its effectiveness.

Answer and Implementation for the Challenge:

	To add multi-language support to the topic-filtering system, we can extend the current filtering mechanism by: **Translating Input into a Common Language** Use a language translation API (e.g., Google Translate API) to translate user input into a common language, such as English. This ensures all input can be processed against a single predefined keyword list. **Expanding the Keyword List** Create keyword lists for each supported language (e.g., Spanish, French, Mandarin). For instance: English: climate, recycling, pollution Spanish: clima, reciclaje, contaminación French: climat, recyclage, pollution Mandarin: 气候, 回收, 污染 **Matching Input Against All Lists** Instead of using a single list for validation, compare the translated input or its original version to all keyword lists. If any match occurs, the input is valid. **WAY TO IMPROVE, What are some ways to improve the input filtering system to make it more accurate and user-friendly?**